STUDY OF SOME FINGERPRINT VERIFICATION ALGORITHMS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

In

Telematics and Signal Processing

By

KOMMU AYYANNA

20507023

Department of Electronics & Communication Engineering

National Institute of Technology

Rourkela

2007

STUDY OF SOME FINGERPRINT VERIFICATION ALGORITHMS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology
In
Telematics and Signal Processing

By
KOMMU AYYANNA
20507023

Under the Guidance of
Prof. S. MEHER

Department of Electronics & Communication Engineering
National Institute of Technology
Rourkela
2007

NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA

CETRIFICATE

This is to certify that the Thesis Report titled **"Study of some Fingerprint Verification Algorithms"** submitted by Mr. **Kommu Ayyanna (20507023)** in partial fulfillment of the requirements for the award of Master of Technology degree Electronics and Communication Engineering with specialization in "Telematics and Signal Processing" during session 2006-2007 at National Institute Of Technology, Rourkela (Deemed University) and is an authentic work by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university/institute for the award of any Degree of Diploma.

Prof. S. Meher
Dept. of E.C.E
National institute of Technology
Rourkela-769008

Date:

Contents

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis guide **Prof S. Meher** for his guidance, advice and constant support throughout my thesis work. I would like to thank him for being my advisor here at National Institute of Technology, Rourkela.

Next, I thank **Prof. G. Panda** (Head of Department) for his constant encouragement and support during my postgraduate studies. I would like to thank **Prof. G. S. Rath, Prof. K. K. Mahapatra,** and **Prof. S.K. Patra** for teaching me and also helping me how to learn. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I would like to thank all faculty members and staff of the Department of Electronics and Communication Engineering, N.I.T. Rourkela for their generous help in various ways for the completion of this thesis.

I would also like to mention the names of **N. Bhoi, R.K. Kulakarni, C.S. Rawat, M. Kamal Kumar, Manoj Kumar Gupta,** and **Nirulata** for helping me a lot during the thesis period.

I would like to thank all my friends and especially my classmates for all the thoughtful and mind stimulating discussions we had, which prompted us to think beyond the obvious. I've enjoyed their companionship so much during my stay at NIT, Rourkela.

Last but not least I would like to thank my parents, who taught me the value of hard work by their own example. They rendered me enormous support during the whole tenure of my stay in NIT Rourkela.

Kommu Ayyanna
Roll No: 20507023
Dept of ECE, NIT, Rourkela

ABSTRACT

Fingerprint Verification is one of the most reliable personal identification methods. However, manual fingerprint verification is so tedious, time-consuming, and expensive that it is incapable of meeting today's increasing performance. An automatic fingerprint identification system (AFIS) is widely needed. It plays a very important role in forensic and civilian applications such as criminal identification, access control, and ATM card verification. Image processing provides power tools for this purpose. The target can be mainly decomposed into image preprocessing, feature extraction and feature match. For each sub-task, some classical and up-to-date methods in literatures are analyzed. Based on the analysis, an integrated solution for fingerprint recognition is implemented.

A new feature extraction algorithm based on Crossing Number (CN) concept is presented. This method involves the use of the skeleton image where the ridge flow pattern is eight-connected. The minutiae are extracted by scanning the local neighbourhood of each ridge pixel in the image using a 3x3 window. The CN value is then computed, which is defined as half the sum of the differences between pairs of adjacent pixels in the eight-neighbourhood. The algorithm has several advantages over the techniques proposed in literature such as increased computational efficiency, improved localization and higher sensitivity.

Further a novel alignment based fingerprint recognition algorithm for minutiae matching is discussed. Finally an extraction method of DWT and DCT features for fingerprint matching is simulated and discussed. The recognition rate of this method is evaluated by the k-NN classifier. This method has the advantage of both higher recognition rate and lower complexity.

List of Figures

List of Tables

Chapter 1

INTRODUCTION

In an increasingly digital world, reliable personal authentication has become an important human computer interface activity. National security, e-commerce, and access to computer networks are some examples where establishing a person's identity is vital. Existing security measures rely on knowledge-based approaches like passwords or token-based approaches such as swipe cards and passports to control access to physical and virtual spaces. Though ubiquitous, such methods are not very secure. Tokens such as badges and access cards may be shared or stolen. Passwords and PIN numbers may be stolen electronically. Furthermore, they cannot differentiate between authorized user and a person having access to the tokens or knowledge.

Biometrics such as fingerprint, face and voice print offers means of reliable personal authentication that can address these problems and is gaining citizen and government acceptance.

1.1 BIOMETRICS

Biometrics is the science of verifying the identity of an individual through physiological measurements or behavioral traits. Since biometric identifiers are associated permanently with the user they are more reliable than token or knowledge based authentication methods. Biometrics offers several advantages over traditional security measures. These include

1. **Non-repudiation**: With token and password based approaches, the perpetrator can always deny committing the crime pleading that his/her password or ID was stolen or compromised even when confronted with an electronic audit trail. There is no way in which his claim can be verified effectively. This is known as the problem of deniability or of 'repudiation'. However, biometrics is indefinitely associated with a user and hence it cannot be lent or stolen making such repudiation infeasible.
2. **Accuracy and Security**: Password based systems are prone to dictionary and brute force attacks. Furthermore, such systems are as vulnerable as their weakest password. On the other hand, biometric authentication requires the physical presence of the user and therefore cannot be circumvented through a dictionary or brute force style attack. Biometrics have also been shown to possess a higher bit strength compared to password based systems [24] and are therefore inherently secure.
3. **Screening**: In screening applications, we are interested in preventing the users from assuming multiple identities (e.g. a terrorist using multiple passports to enter a foreign

country). This requires that we ensure a person has not already enrolled under another assumed identity before adding his new record into the database. Such screening is not possible using traditional authentication mechanisms and biometrics provides the only available solution.

The various biometric modalities can be broadly categorized as

• **Physical biometrics**: This involves some form of physical measurement and includes modalities such as face, fingerprints, iris-scans, hand geometry etc.

• **Behavioral biometrics**: These are usually temporal in nature and involve measuring the *way* in which a user performs certain tasks. This includes modalities such as speech, signature, gait, keystroke dynamics etc.

• **Chemical biometrics**: This is still a nascent field and involves measuring chemical cues such as odor and the chemical composition of human perspiration.

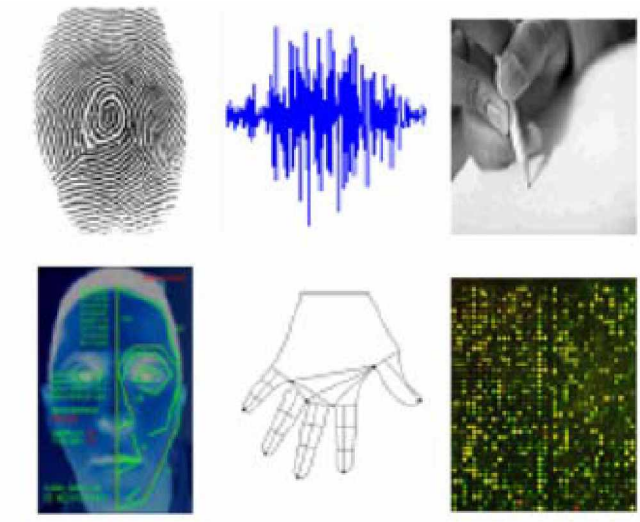


Figure 1.1: Various biometric modalities: Fingerprints, speech, handwriting, face, hand geometry and chemical biometrics

It is also instructive to compare the relative merits and de-merits of biometric and password/cryptographic key based systems. Table 1.1 provides a summary of them.

Depending on the application, biometrics can be used for identification or for verification. In verification, the biometric is used to validate the claim made by the individual. The biometric of the user is compared with the biometric of the claimed individual in the database. The claim is rejected or accepted based on the match. (In essence, the system tries to answer the question,"Am I whom I claim to be?"). In identification, the system recognizes

an individual by comparing his biometrics with every record in the database. (In essence, the system tries to answer the question,

Biometric Authentication	**Password/Key based authentication**
Based on physiological measurements or behavioral traits	Based on something that the user 'has' or 'knows'
Authenticates the user	Authenticates the password/key
Is permanently associated with the user	Can be lent, lost or stolen
Biometric templates have high uncertainty	Have zero uncertainty
Utilizes probabilistic matching	Requires exact match for authentication
Utilizes probabilistic matching	Requires exact match for authentication

Table1.1: Comparison of Biometric and Password/Key based authentication

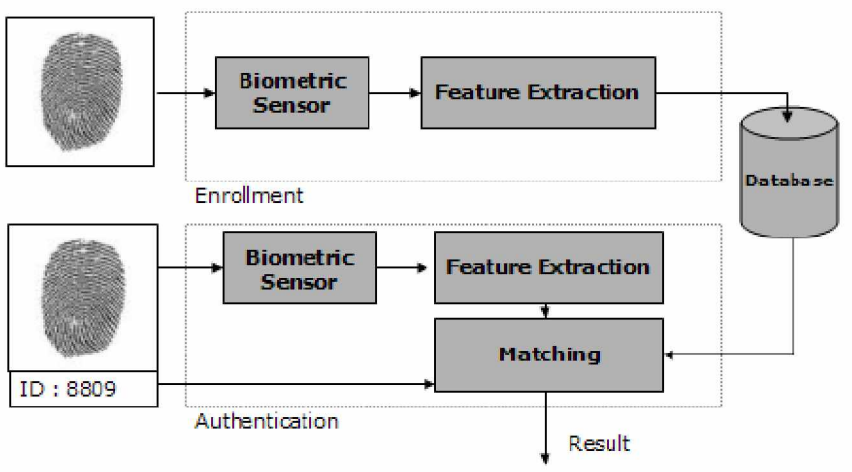


Figure 1.2: General architecture of a biometric system

"Who am I?"). In this thesis, we will be dealing mainly with the problem of verification using fingerprints. In general, biometric verification consists of two stages (Figure 1.2) (i) Enrollment and (ii) Authentication. During enrollment, the biometrics of the user is captured and the extracted features (template) are stored in the database. During authentication, the biometrics of the user is captured again and the extracted features are compared with the ones already existing in the database to determine a match. The specific record to fetch from the database is determined using the claimed identity of the user. The database itself may be central or distributed with each user carrying his template on a smart card.

1.1.1 Biometrics and Pattern Recognition

As recently as a decade ago, biometrics did not exist as a separate field. It has evolved through interaction and confluence of several fields. Fingerprint recognition emerged from the application of pattern recognition to forensics. Speaker verification evolved out of the signal processing community. Face detection and recognition was largely researched by the computer vision community. While biometrics is primarily considered as application of pattern recognition techniques, it has several outstanding differences from conventional classification problems as enumerated below

1. In a conventional pattern classification problem such as Optical Character Recognition (OCR) recognition, the number of patterns to classify is small (A-Z) compared to the number of samples available for each class. However in case of biometric recognition, the number of classes is as large as the set of individuals in the database. Moreover, it is very common that only a single template is registered per user.
2. The primary task in biometric recognition is that of choosing a proper feature representation. Once the features are carefully chosen, the act of performing verification is fairly straightforward and commonly employs simple metrics such as Euclidean distance. Hence the most challenging aspects of biometric identification involves signal and image processing for feature extraction.
3. Since biometric templates represent personally identifiable information of individuals, security and privacy of the data is of particular importance unlike other applications of pattern recognition.
4. Modalities such as fingerprints, where the template is expressed as an unordered point set (minutiae) do not fall under the category of traditional multi-variate/vectorial features commonly used in pattern recognition.

1.1.2 The Verification Problem

Here we consider the problem of biometric verification in a more formal manner. In a verification problem, the biometric signal from the user is compared against a single enrolled template. This template is chosen based on the claimed identity of the user. Each user i is represented by a biometric B_i. It is assumed that there is a one-to-one correspondence between the biometric B_i and the identity i of the individual. The feature extraction phase results in a machine representation (template) T_i of the biometric.

During verification, the user claims an identity j and provides a biometric signal B_j. The feature extractor now derives the corresponding machine representation T_j. The recognition consists of computing a similarity score S (T_i, T_j). The claimed identity is assumed to be true if the S (T_i, T_j) > Th for some threshold Th The choice of the threshold also determines the trade-off between user convenience and system security as will be seen in the ensuing section.

Figure 1.3: An illustration showing the intra-user variation present in biometric signals

1.1.3 Performance Verification

Unlike passwords and cryptographic keys, biometric templates have high uncertainty. There is considerable variation between biometric samples of the same user taken at different instances of time (Figure 1.3). Therefore the match is always done probabilistically. This is in contrast to exact match required by password and token based approaches. The inexact matching leads to two forms of errors

• **False Accept** An impostor may sometime be accepted as a genuine user, if the similarity with his template falls within the intra-user variation of the genuine user.

• **False Reject** When the acquired biometric signal is of poor quality, even a genuine user may be rejected during authentication. This form of error is labeled as a 'false reject'.

The system may also have other less frequent forms of errors such as

• **Failure to enroll (FTE)** It is estimated that nearly 4% of the population have illegible fingerprints. This consists of senior population, laborers who use their hands a lot and injured individuals. Due to the poor ridge structure present in such individuals, such users cannot be enrolled into the database and therefore cannot be subsequently authenticated. Such individuals are termed as 'goats' [1]. A biometric system should have exception handling mechanism in place to deal with such scenarios.

• **Failure to authenticate (FTA)** This error occurs when the system is unable to extract features during verification even though the biometric was legible during enrollment. In case of fingerprints this may be caused due to excessive sweating, recent injury etc. In case of speech, this may be caused due to cold, sore throat etc. It should be noted that this error is distinct from False Reject where the rejection occurs during the matching phase. In FTA, the rejection occurs in the feature extraction stage itself.

1.2 FINGERPRINTS AS BIOMETRIC

Fingerprints were accepted formally as valid personal identifier in the early twentieth century and have since then become a de-facto authentication technique in law-enforcement agencies world wide. The FBI currently maintains more than 400 million fingerprint records on file. Fingerprints have several advantages over other biometrics, such as the following:

1. **High universality**: A large majority of the human population has legible fingerprints and can therefore be easily authenticated. This exceeds the extent of the population who possess passports, ID cards or any other form of tokens.

2. **High distinctiveness**: Even identical twins who share the same DNA have been shown to have different fingerprints, since the ridge structure on the finger is not encoded in the genes of an individual. Thus, fingerprints represent a stronger authentication mechanism than DNA. Furthermore, there has been no evidence of identical fingerprints in more than a century of forensic practice. There are also mathematical models [3] that justify the high distinctiveness of fingerprint patterns.

3. **High permanence**: The ridge patterns on the surface of the finger are formed in the womb and remain invariant until death except in the case of severe burns or deep physical injuries.

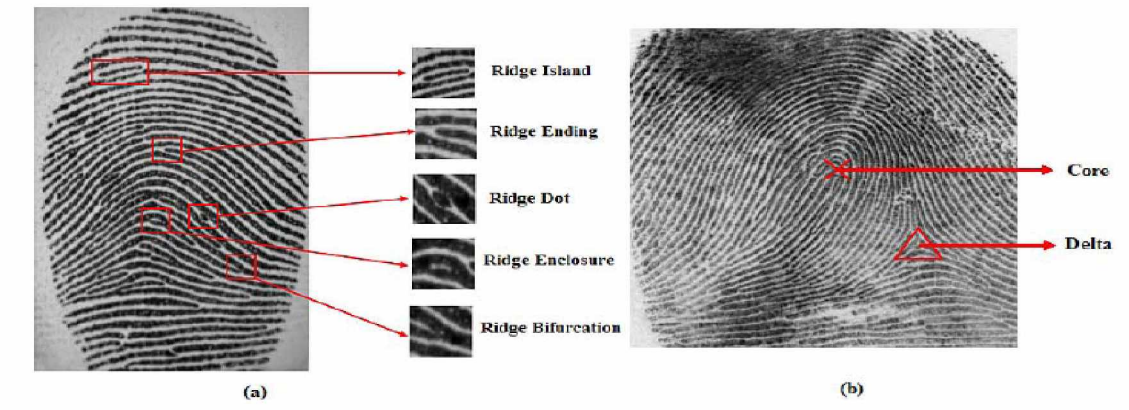


Figure 1.4: (a) Local Features: Minutiae (b) Global Features: Core and Delta

4. Easy collectability: The process of collecting fingerprints has become very easy with the advent of online sensors. These sensors are capable of capturing high resolution images of the finger surface within a matter of seconds [2]. This process requires minimal or no user training and can be collected easily from co-operative or non co-operative users. In contrast, other accurate modalities like iris recognition require very co-operative users and have considerable learning curve in using the identification system.

5. **High performance**: Fingerprints remain one of the most accurate biometric modalities available to date with jointly optimal FAR (false accept rate) and FRR (false reject rate). Forensic systems are currently capable of achieving FAR of less than 10^{-4}.

6. **Wide acceptability**: While a minority of the user population is reluctant to give their fingerprints due to the association with criminal and forensic fingerprint databases, it is by far the most widely used modality for biometric authentication.

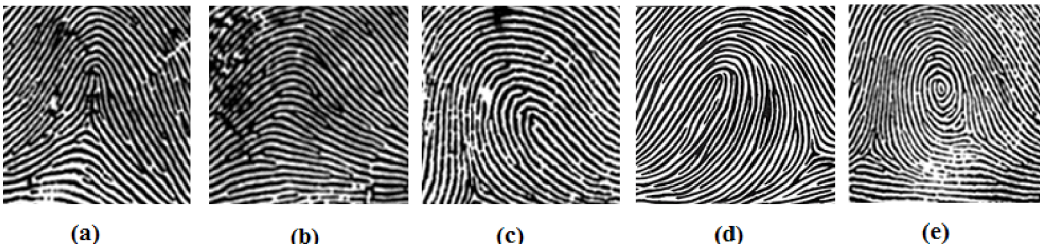


Figure 1.5 Fingerprint Classes:(a)Tented Arch(b)Arch(c)Right Loop(d)Left Loop(e)Whorl

The fingerprint surface is made up of a system of ridges and valleys that serve as friction surface when we are gripping the objects. The surface exhibits very rich structural information when examined as an image. The fingerprint images can be represented by both global as well as local features. The global features include the ridge orientation, ridge spacing and singular points such as core and delta. The singular points are very useful from the classification perspective (See Figure 1.5). However, verification usually relies exclusively on minutiae features. Minutiae are local features marked by ridge discontinuities. There are about 18 distinct types of minutiae features that include ridge endings, bifurcations, crossovers and islands. Among these, *ridge endings* and *bifurcation* are the commonly used features (See Figure 1.4). A *ridge ending* occurs when the ridge flow abruptly terminates and a *ridge bifurcation* is marked by a fork in the ridge flow. Most matching algorithms do not even differentiate between these two types since they can easily get exchanged under

different pressures during acquisition. Global features do not have sufficient discriminative power on their own and are therefore used for binning or classification before the extraction of the local minutiae features.

The various stages of a typical fingerprint recognition system is shown in Figure 1.6. The fingerprint image is acquired using off-line methods such as creating an inked impression on paper or through a live capture device consisting of an optical, capacitive, ultrasound or thermal sensor [2]. The first stage consists of standard image processing algorithms such as noise removal and smoothening. However, it is to be noted that unlike regular images, the fingerprint

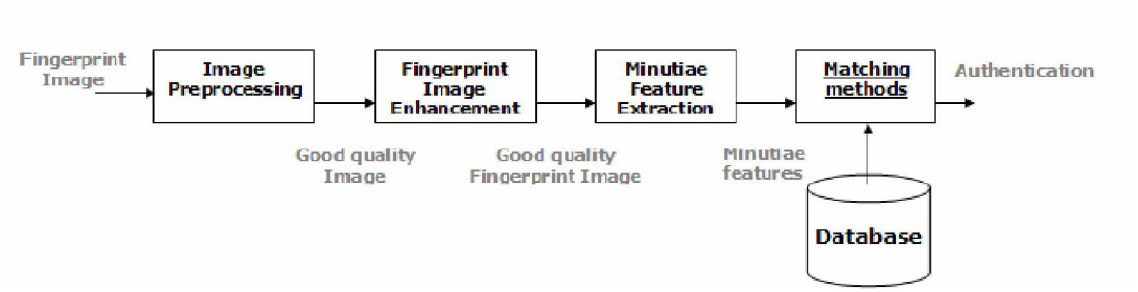


Figure 1.6: General architecture of a fingerprint verification system

image represents a system of oriented texture and has very rich structural information within the image. Furthermore, the definition of noise and unwanted artifacts are also specific to fingerprints. The fingerprint image enhancement algorithms are specifically designed to exploit the periodic and directional nature of the ridges. Finally, the minutiae features are extracted from the image and are subsequently used for matching. Although research in fingerprint verification research has been pursued for several decades now, there are several open research challenges still remaining, some of which will be addressed in the ensuing sections of this thesis.

1.3 APPLICATIONS OF FINGERPRINT RECOGNITION SYSTEMS

Fingerprint recognition is a rapidly evolving technology that has been widely used in forensics such as criminal recognition and prison security, and has a very strong potential to be widely adopted in a broad range of civilian applications (see Table 1.2 and Figure 1.7).

Forensic	Government	Commercial
Corpse Identification, Criminal Investigation, Terrorist Identification, Parenthood Determination, Missing Children, etc.	National ID card, Correctional Facility, Driver's License, Social Security, Welfare Disbursement, Border Control, Passport Control, etc.	Computer Network Logon, Electronic Data Security, E-Commerce, Internet Access, ATM, Credit Card, Physical Access Control, Cellular Phones, Personal Digital Assistant, Medical Records Management Distance Learning, etc.

Table1.2. Most of the fingerprint recognition applications are divided here into three categories.

Traditionally, forensic applications have used manual biometrics, government applications have used token-based systems, and commercial applications have used knowledge-based systems. Fingerprint recognition systems are now being increasingly used for all these sectors. Note that over one billion dollars in welfare benefits are annually claimed by "double dipping" welfare recipients in the United States alone.

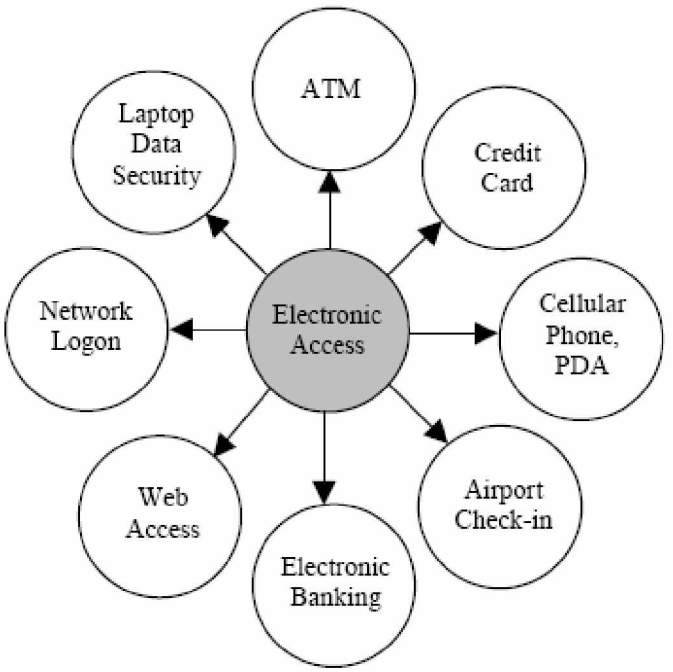


Figure1.7. Various electronic access applications in widespread use that require automatic recognition.

These applications may be divided into the following groups: i) applications such as banking, electronic commerce, and access control, in which biometrics will replace or enforce the current token- or knowledge-based techniques and ii) applications such as welfare and immigration in which neither the token-based nor the knowledge-based techniques are currently being used.

Information system/computer network security, such as user authentication and access to databases via remote login, is one of the most important application areas for fingerprint recognition. It is expected that more and more information systems/computer networks will be secured with fingerprints with the rapid expansion of the Internet. Applications such as medical information systems, distance learning and e-publishing are already benefiting from deployment of such systems. *Electronic commerce and electronic banking* are also important and emerging application areas of biometrics due to the rapid progress in electronic transactions. These applications include electronic fund transfers, ATM security, check cashing, credit card security, smartcard security, on-line transactions, and so on. Currently, there are several large fingerprint security projects under development in these areas, including credit card security (MasterCard) and smartcard security (IBM and American Express).

The *physical access control* market is currently dominated by token-based technology. However, it is increasingly shifting to fingerprint-based biometric techniques. The introduction of fingerprint-based biometrics in *government benefits distribution programs* such as welfare disbursement has already resulted in substantial savings in deterring multiple claimants. In addition, *customs and immigration initiatives* such as the *INS* Passenger Accelerated Service System (INSPASS) which permits faster immigration procedures based on hand geometry will greatly increase operational efficiency. Fingerprint-based *national ID systems* provide a unique ID to the citizens and integrate different government services. Fingerprint-based *voter and driver registration* provides registration facilities for voters and drivers. Fingerprint-based *time/attendance monitoring systems* can be used to prevent any abuses of the current token based/ manual systems. Fingerprint-based recognition systems will replace passwords and tokens in a large number of applications. Their use will increasingly reduce identity theft and fraud and protect privacy.

As fingerprint technology matures, there will be increasing interaction among market, technology, and applications. The emerging interaction is expected to be influenced by the added value of the technology, the sensitivities of the user population, and the credibility of the service provider. It is too early to predict where and how fingerprint technology would evolve and be mated with which applications, but it is certain that fingerprint-based recognition will have a profound influence on the way we will conduct our daily business.

1.4 ORGANIZATION OF THESIS

Following the introduction, the rest of the thesis is organized as follows. Chapter 2 gives a review on existing techniques in the field of fingerprint image enhancement, feature extraction and matching. Chapter 3 introduces a fingerprint image enhancement algorithm based on Gabor filtering. Chapter 4 presents a novel feature extraction algorithm based on crossing number (CN) concept. Chapter 5 describes an alignment-based minutiae matching algorithm. We present a novel DWT and DCT based matching algorithm for matching fingerprints in Chapter 6. Then I made a conclusion to my work and the points to possible directions for future work in Chapter 7.

Chapter 2

LITERATURE REVIEW

2.1 FINGERPRINT IMAGE ENHANCEMENT

One of the most widely cited fingerprint enhancement techniques is the method employed by Hong et al. [4], which is based on the convolution of the image with Gabor filters tuned to the local ridge orientation and ridge frequency. The main stages of this algorithm include normalization, ridge orientation estimation, ridge frequency estimation and filtering.

The first step in this approach involves the normalization of the fingerprint image so that it has a prespecified mean and variance. Due to imperfections in the fingerprint image capture process such as non-uniform ink intensity or non-uniform contact with the fingerprint capture device, a fingerprint image may exhibit distorted levels of variation in grey-level values along the ridges and valleys. Thus, normalization is used to reduce the effect of these variations, which facilitates the subsequent image enhancement steps.

An orientation image is then calculated, which is a matrix of direction vectors representing the ridge orientation at each location in the image. The widely employed gradient-based approach is used to calculate the gradient [6], which makes use of the fact that the orientation vector is orthogonal to the gradient. Firstly, the image is partitioned into square blocks and the gradient is calculated for every pixel, in the x and y directions. The orientation vector for each block can then be derived by performing an averaging operation on all the vectors orthogonal to the gradient pixels in the block. Due to the presence of noise and corrupted elements in the image, the ridge orientation may not always be correctly determined. Given that the ridge orientation varies slowly in a local neighbourhood, the orientation image is then smoothed using a low-pass filter to reduce the effect of outliers.

The next step in the image enhancement process is the estimation of the ridge frequency image. The frequency image defines the local frequency of the ridges contained in the fingerprint. Firstly, the image is divided into square blocks and an oriented window is calculated for each block. For each block, an x-signature signal is constructed using the ridges and valleys in the oriented window. The x-signature is the projection of all the grey level values in the oriented window along a direction orthogonal to the ridge orientation.

Consequently, the projection forms a sinusoidal-shape wave in which the centre of a ridge maps itself as a local minimum in the projected wave. The distance between consecutive peaks in the x-signature can then be used to estimate the frequency of the ridges.

Fingerprint enhancement methods based on the Gabor filter have been widely used to facilitate various fingerprint applications such as fingerprint matching [7] and fingerprint classification [8]. Gabor filters are band pass filters that have both frequency-selective and orientation-selective properties, which means the filters can be effectively tuned to specific frequency and orientation values. One useful characteristic of fingerprints is that they are known to have well defined local ridge orientation and ridge frequency. Therefore, the enhancement algorithm takes advantage of this regularity of spatial structure by applying Gabor filters that are tuned to match the local ridge orientation and frequency. Based on the local orientation and ridge frequency around each pixel, the Gabor filter is applied to each pixel location in the image. The effect is that the filter enhances the ridges oriented in the direction of the local orientation, and decreases anything oriented differently. Hence, the filter increases the contrast between the foreground ridges and the background, whilst effectively reducing noise.

An alternative approach to enhancing the features in a fingerprint image is the technique employed by Sherlock [9] called directional Fourier filtering. The previous approach was a spatial domain technique that involves spatial convolution of the image with filters, which can be computationally expensive. Alternatively, operating in the frequency domain allows one to efficiently convolve the fingerprint image with filters of full image size.

The image enhancement process begins by firstly computing the orientation image. In contrast to the previous method, which estimates the ridge orientation using a continuous range of directions, this method uses a set of only 16 directions to calculate the orientation. An image window is centred at a point in the raw image, which is used to obtain a projection of the local ridge information. The image window is then rotated in each of the 16 equally spaced directions, and in each direction a projection along the window's *y* axis is formed. The projection with the maximum variance is used as the dominant orientation for that point in the image. This process is then repeated for each pixel to form the orientation image.

Similar to the filtering stage applied by Hong et al.: after the orientation image has been computed, the raw image is then filtered using a set of band pass filters tuned to match the ridge orientation. The image is firstly converted from the spatial domain into the frequency domain by application of the two-dimensional discrete Fourier transform. The Fourier image

is then filtered using a set of 16 Butterworth filters with each filter tuned to a particular orientation. The number of directional filters corresponds to the set of directions used to calculate the orientation image. After each directional filter has been independently applied to the Fourier image, the inverse Fourier transform is used to convert each image back to the spatial domain, thereby producing a set of directionally filtered images called prefiltered images.

The next step in the enhancement process is to construct the final filtered image using the pixel values from the prefiltered images. This requires the value of the ridge orientation at each pixel in the raw image and the filtering direction of each prefiltered image. Each point in the final image is then computed by selecting, from the prefiltered images the pixel value whose filtering direction most closely matches the actual ridge orientation. The output of the filtering stage is an enhanced version of the image that has been smoothed in the direction of the ridges.

Lastly, local adaptive thresholding is applied to the directionally filtered image, which produces the final enhanced binary image. This involves calculating the average of the grey-level values within an image window at each pixel, and if the average is greater than the threshold, then the pixel value is set to a binary value of one; otherwise, it is set to zero. The grey-level image is converted to a binary image, as there are only two levels of interest, the foreground ridges and the background valleys.

Overall, it can be seen that most techniques for fingerprint image enhancement are based on filters that are tuned according to the local characteristics of fingerprint images. Both of the examined techniques employ the ridge orientation information for tuning of the filter. However, only the approach by Hong et al. [4] takes into account the ridge frequency information, as Sherlock's approach assumes the ridge frequency to be constant. By using both the orientation and ridge frequency information, it allows for accurate tuning of the Gabor filter parameters, which consequently leads to better enhancement results. Hence, I have chosen to employ the Gabor filtering approach by Hong et al. [4] to perform fingerprint image enhancement.

2.2 MINUTIAE EXTRACTION AND POSTPROCESSING

After a fingerprint image has been enhanced, the next step is to extract the minutiae from the enhanced image. Following the extraction of minutiae, a final image postprocessing stage is

performed to eliminate false minutiae. This section contains a review of existing literature in the field of minutiae extraction and postprocessing.

2.2.1 Minutiae Extraction

The most commonly employed method of minutiae extraction is the Crossing Number (CN) concept [10]. This method involves the use of the skeleton image where the ridge flow pattern is eight-connected. The minutiae are extracted by scanning the local neighbourhood of each ridge pixel in the image using a 3x3 window. The CN value is then computed, which is defined as half the sum of the differences between pairs of adjacent pixels in the eight-neighbourhood. Using the properties of the CN as shown in Table 3.1, the ridge pixel can then be classified as a ridge ending, bifurcation or non-minutiae point. For example, a ridge pixel with a CN of one corresponds to a ridge ending, and a CN of three corresponds to a bifurcation.

CN	**Property**
0	Isolated point
1	Ridge ending point
2	Continuing ridge ending point
3	Bifurcation point
4	Crossing point

Table 2.1: Properties of the Crossing Number.

Other authors such as Jain et al [5].and Ratha et al. [6] have also performed minutiae extraction using the skeleton image. Their approach involves using a 3×3 window to examine the local neighbourhood of each ridge pixel in the image. A pixel is then classified as a ridge ending if it has only one neighbouring ridge pixel in the window, and classified as a bifurcation if it has three neighbouring ridge pixels. Consequently, it can be seen that this approach is very similar to the Crossing Number method.

2.2.2 Fingerprint Image Postprocessing

False minutiae may be introduced into the image due to factors such as noisy images, and image artifacts created by the thinning process. Hence, after the minutiae are extracted, it is necessary to employ a postprocessing stage in order to validate the minutiae. Figure 2.1 illustrates some examples of false minutiae structures, which include the spur, hole, triangle and spike structures. It can be seen that the spur structure generates false ridge endings; where as both the hole and triangle structures generate false bifurcations. The spike structure creates a false bifurcation and a false ridge ending point.

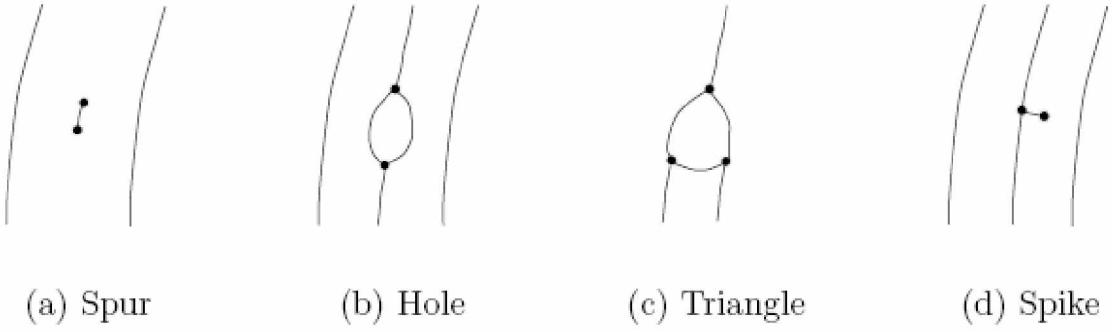


Figure 2.1: Examples of typical false minutiae structures.

The majority of the proposed approaches for image postprocessing in literature are based on a series of structural rules used to eliminate spurious minutiae. One such approach is the one proposed by Ratha et al. [6], which performs the validation of minutiae based on a set of heuristic rules. For example, a ridge ending point that is connected to a bifurcation point, and is below a certain threshold distance is eliminated. This heuristic rule corresponds to removal of the spike structure shown in Figure 2.1(d). Additional heuristic rules are then used to eliminate other types of false minutiae. Furthermore, a boundary effect treatment is applied where the minutiae below a certain distance from the boundary of the foreground region are deleted.

A novel approach to the validation of minutiae is the postprocessing algorithm proposed by Tico and Kuosmanen [11]. Similar to the above techniques, this algorithm operates on the skeleton image. However, rather than employing a different set of heuristics each time to eliminate a specific type of false minutiae, this approach incorporates the validation of different types of minutiae into a single algorithm. It tests the validity of each minutiae point by scanning the skeleton image and examining the local neighbourhood around the minutiae. The algorithm is then able to cancel out false minutiae based on the configuration of the ridge pixels connected to the minutiae point.

2.3. FINGERPRINT MATCHING

Clearly the most important stage of a fingerprint verification system is the matching process. The matching algorithm takes template representations (T, I) of two different fingerprints and returns a similarity score S (T, I). The representation T and I may either be image, texture descriptors or minutiae information as outlined in Section 1. In this thesis, our primary focus will be on the minutiae representation.

2.3.1 Alignment-based Matching Algorithm

Generally, an automatic fingerprint verification/identification is achieved with point pattern matching (minutiae matching) instead of a pixel-wise matching or a ridge pattern matching of fingerprint images. A number of point pattern matching algorithms have been proposed in the literature [12], [13]. Because a general point matching problem is essentially intractable, features associated with each point and their spatial properties such as the relative distances between points are often used in these algorithms to reduce the exponential number of search paths.

The relaxation approach [13] iteratively adjusts the confidence level of each corresponding pair based on its consistency with other pairs until a certain criterion is satisfied. Although a number of modified versions of this algorithm have been proposed to reduce the matching complexity, these algorithms are inherently slow because of their iterative nature.

The Hough transform-based approach proposed by Stockman et al. converts point pattern matching to a problem of detecting the highest peak in the Hough space of transformation parameters. It discretizes the transformation parameter space and accumulates evidence in the discretized space by deriving transformation parameters that relate two point patterns using a substructure or feature matching technique. Karu and Jain [14] proposed a hierarchical Hough transform-based registration algorithm which greatly reduced the size of accumulator array by a multiresolution approach. However, if the number of minutia point is less than 30, then it is very difficult to accumulate enough evidence in the Hough transform space for a reliable match.

Another approach to point matching is based on energy minimization. This approach defines a cost function based on an initial set of possible correspondences and uses an appropriate optimization algorithm such as genetic algorithm and simulated annealing [12] to find a possible suboptimal match. These methods tend to be very slow and are unsuitable for fingerprint verification system.

In our system, an alignment-based matching algorithm is implemented. Recognition by alignment has received a great deal of attention during the past few years, because it is simple in theory, efficient in discrimination, and fast in speed. Our alignment-based matching algorithm decomposes the minutia matching into two stages:

1) *Alignment stage*, where transformations such as translation, rotation and scaling between an input and a template in the database are estimated and the input minutiae are aligned with the template minutiae according to the estimated parameters; and

2) *Matching stage*, where both the input minutiae and the template minutiae are converted to polygons in the polar coordinate system and an elastic string matching algorithm is used to match the resulting polygons.

2.3.2 DWT and DCT Feature-based Fingerprint Matching

Various algorithms for fingerprint matching have been proposed in the literature. Most of them are quite complex and require large amount of processing time, which is rather impractical for wireless applications. Basically, fingerprint matching method is categorized in three classes [2], namely, correlation-based matching, minutiae-based matching and ridge feature-based matching. The correlation-based matching is a straight method used to compare the corresponding pixels of the fingerprint images in many of rotating and shifting. For the minutiae-based method, e.g. [17] and [18], the fingerprint information called *minutiae* is extracted from the fingerprint image and used in the matching process. However, this method normally requires some pre-processing processes called *binarization* and *thinning*. Binarization is a process used to change a gray-scale image to a black-and-white one, while thinning is a process used to change a thick ridge line to the one that has only one pixel width. Ultimately, the objectives of the pre-processing process are the image normalization and background noise elimination. For example, in fingerprint minutiae classification and recognition [17] and minutiae detection algorithm [18], the minutiae called ridge termination and ridge bifurcation were extracted

from the fingerprint image. A ridge termination is defined as the ridge point, where a ridge ends suddenly, while a ridge bifurcation is defined as the ridge point, where a ridge forks or diverges into branch ridges. Usually, minutiae are extracted from two fingerprint images and saved as sets of points in two-dimensional plane. The sets of points are then used to find matching points of two fingerprint images that give the maximum number of minutiae pairings.

In ridge feature-based matching, the feature of ridge is extracted from the gray-scale image, and represented by other parameters. Those parameters, depending on the tools used in the feature extraction process, are then used for matching fingerprints. The GABOR filter-based method [19, 20] is an example of ridge feature-based matching. Accordingly, it extracts the feature directly from the fingerprint image in 4 orientations (0°, 45°, 90° and 145°) using the GABOR filter. The coefficients obtained are then used as a matching parameter. This approach gains advantage from less preprocessing effort than minutiae-based method, and is hence possible to be implemented in wireless pervasive computing environment. Using the similar approach, Tico *et al* [21, 22] demonstrated later that, by using the discrete wavelet transform (DWT) for feature extraction, a higher recognition rate can be obtained. In their scheme, the fingerprint features are generated from three detail sub-bands of DWT coefficients at 4 octaves decomposition, and the standard deviation [21] and norm of each wavelet sub-image [22] are used as the matching parameters.

Chapter 3

FINGERPRINT IMAGE ENHANCEMENT

3.1 INTRODUCTION

The quality of the ridge structures in a fingerprint image is an important characteristic, as the ridges carry the information of characteristic features required for minutiae extraction. Ideally, in a well-defined fingerprint image, the ridges and valleys should alternate and flow in locally constant direction. This regularity facilitates the detection of ridges and consequently, allows minutiae to be precisely extracted from the thinned ridges. However, in practice, a fingerprint image may not always be well defined due to elements of noise that corrupt the clarity of the ridge structures. This corruption may occur due to variations in skin and impression conditions such as scars, humidity, dirt, and non-uniform contact with the fingerprint capture device. Thus, image enhancement techniques are often employed to reduce the noise and enhance the definition of ridges against valleys.

This chapter provides discussion on the methodology and implementation of a fingerprint image enhancement algorithm. The results of the experiments involving each stage of the fingerprint enhancement algorithm will then be presented.

3.2 ALGORITHM FOR FINGERPRINT IMAGE ENHANCEMENT

This section describes the methods for constructing a series of image enhancement techniques for fingerprint images. The algorithm consists of the following stages:

- segmentation,
- normalization,
- orientation estimation,
- ridge frequency estimation,
- Gabor filtering.
- binarization, and
- thinning.

In this section, I will discuss the methodology for each stage of the enhancement algorithm, including any modifications that have been made to the original techniques.

3.2.1 Segmentation

The first step of the fingerprint enhancement algorithm is image segmentation. Segmentation is the process of separating the foreground regions in the image from the background regions. The foreground regions correspond to the clear fingerprint area containing the ridges and

valleys, which is the area of interest. The background corresponds to the regions outside the borders of the fingerprint area, which do not contain any valid fingerprint information. When minutiae extraction algorithms are applied to the background regions of an image, it results in the extraction of noisy and false minutiae. Thus, segmentation is employed to discard these background regions, which facilitates the reliable extraction of minutiae.

In a fingerprint image, the background regions generally exhibit a very low grey-scale variance value, whereas the foreground regions have a very high variance. Hence, a method based on variance thresholding [10] can be used to perform the segmentation. Firstly, the image is divided into blocks and the grey-scale variance is calculated for each block in the image. If the variance is less than the global threshold, then the block is assigned to be a background region; otherwise, it is assigned to be part of the foreground. The grey-level variance for a block of size *W x W* is defined as:

$$V(k) = \frac{1}{w^2} \sum_{i=0}^{w-1} \sum_{j=0}^{w-1} (I(i, j) - M(k))^2, \tag{3.1}$$

Where $V(k)$ is the variance for block k, I (i, j) is the grey-level value at pixel (i, j), and $M(k)$ is the mean grey-level value for the block k.

3.2.2 Normalization

The next step in the fingerprint enhancement process is image normalization. Normalization is used to standardize the intensity values in an image by adjusting the range of grey-level values so that it lies within a desired range of values. Let $I(i, j)$ represent the grey-level value at pixel (i, j), and $N(i, j)$ represent the normalized grey-level value at pixel (i, j). The normalized image is defined as:

$$N(i, j) = \begin{cases} M_0 + \sqrt{\dfrac{V_0 (I(i, j) - M)^2}{V}}, & \text{if } I(i, j) > M \\ M_0 - \sqrt{\dfrac{V_0 (I(i, j) - M)^2}{V}}, & \text{otherwise} \end{cases} \tag{3.2}$$

Where M and V are the estimated mean and variance of I(i, j), respectively, and M_0 and V_0 are the desired mean and variance values, respectively. Normalization does not change the ridge structures in a fingerprint; it is performed to standardize the dynamic levels of variation in grey-level values, which facilitates the processing of subsequent image enhancement stages.

3.2.3 Orientation estimation

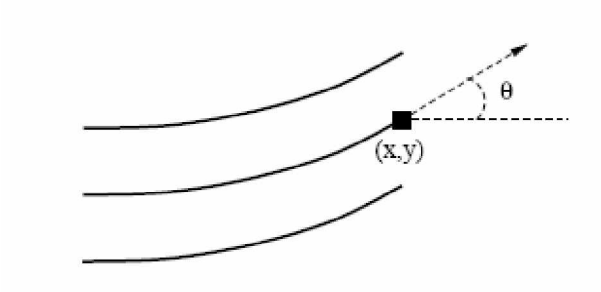


Figure 3.1: The orientation of a ridge pixel in a fingerprint.

The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint (see Figure 3.1). The orientation estimation is a fundamental step in the enhancement process as the subsequent Gabor filtering stage relies on the local orientation in order to effectively enhance the fingerprint image. The least mean square estimation method employed by Hong et al. [4] is used to compute the orientation image. However, instead of estimating the orientation block-wise, I have chosen to extend their method into a pixel-wise scheme, which produces a finer and more accurate estimation of the orientation field. The steps for calculating the orientation at pixel (i, j) are as follows:

1. Firstly, a block of size $W \times W$ is centred at pixel (i, j) in the normalized fingerprint image.
2. For each pixel in the block, compute the gradients $\partial x(i, j)$ and $\partial y(i, j)$, which are the gradient magnitudes in the x and y directions, respectively. The horizontal Sobel operator is used to compute $\partial x(i, j)$:

$$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \qquad (3.3)$$

The vertical Sobel operator is used to compute $\partial y(i, j)$:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \qquad (3.4)$$

3. The local orientation at pixel (i, j) can then be estimated using the following

$$v_x(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v), \tag{3.5}$$

$$v_y(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (\partial^2{}_x(u,v)\partial^2{}_y(u,v)), \tag{3.6}$$

$$\theta(i, j) = \frac{1}{2}\tan^{-1}\left[\frac{v_x(i, j)}{v_y(i, j)}\right], \tag{3.7}$$

where $\theta(i, j)$ is the least square estimate of the local orientation at the block centred at pixel (i, j).

4. Smooth the orientation field in a local neighbourhood using a Gaussian filter. The orientation image is firstly converted into a continuous vector field, which is defined as:

$$\phi_x(i, j) = \cos(2\theta(i, j)), \tag{3.8}$$

$$\varphi_y(i, j) = \sin(2\theta(i, j)), \tag{3.9}$$

where ϕ_x and ϕ_y are the x and y components of the vector field, respectively. After the vector field has been computed, Gaussian smoothing is then performed as follows:

$$\phi_x'(i, j) = \sum_{u=-w_{\phi/2}}^{w_{\phi/2}} \sum_{v=-w_{\phi/2}}^{w_{\phi/2}} G(u,v)\phi_x(i-uw, j-vw), \tag{3.10}$$

$$\phi_y'(i, j) = \sum_{u=-w_{\phi/2}}^{w_{\phi/2}} \sum_{v=-w_{\phi/2}}^{w_{\phi/2}} G(u,v)\phi_y(i-uw, j-vw), \tag{3.11}$$

where G is a Gaussian low-pass filter of size $w_x \times w_y$.

5. The final smoothed orientation field O at pixel (i, j) is defined as:

$$O(i, j) = \frac{1}{2}\tan^{-1}\left(\frac{\phi_y'(i, j)}{\phi_x'(i, j)}\right) \tag{3.12}$$

3.2.4 Ridge frequency estimation

In addition to the orientation image, another important parameter that is used in the construction of the Gabor filter is the local ridge frequency. The frequency image represents the local frequency of the ridges in a fingerprint. The first step in the frequency estimation stage is to divide the image into blocks of size $W \times W$. The next step is to project the grey-level values of all the pixels located inside each block along a direction orthogonal to the local ridge orientation. This projection forms an almost sinusoidal-shape wave with the local minimum points corresponding to the ridges in the fingerprint. An example of a projected waveform is shown in Figure 3.2.

I have modified the original frequency estimation stage used by Hong et al. [5] to include an additional projection smoothing step prior to computing the ridge spacing. This involves smoothing the projected waveform using a Gaussian low pass filter of size $w \times w$ to reduce the effect of noise in the projection. The ridge spacing $S(i, j)$ is then computed by counting the median number of pixels between consecutive minima points in the projected waveform. Hence, the ridge frequency $F(i, j)$ for a block centred at pixel (i, j) is defined as:

$$F(i, j) = \frac{1}{S(i, j)} \tag{3.13}$$

Given that the fingerprint is scanned at a fixed resolution, then ideally the ridge frequency values should lie within a certain range. However, there are cases where a valid frequency value cannot be reliably obtained from the projection. Examples are when no consecutive peaks can be detected from the projection, and also when minutiae points appear in the block. For the blocks where minutiae points appear, the projected waveform does not produce a well-defined sinusoidal shape wave, which can lead to an inaccurate estimation of the ridge frequency. Thus, the out of range frequency values are interpolated using values from neighbouring blocks that have a well-defined frequency.

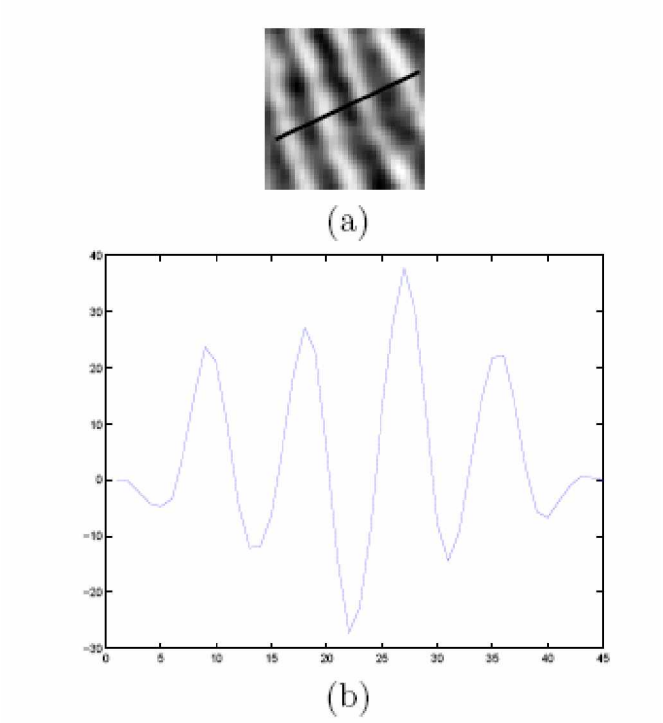


Figure 3.2: The projection of the intensity values of the pixels along a direction orthogonal to the local ridge orientation. (a) A 32 x 32 block from a fingerprint image.
(b) The projected waveform of the block.

3.2.5 Gabor filtering

Once the ridge orientation and ridge frequency information has been determined, these parameters are used to construct the even-symmetric Gabor filter. A two dimensional Gabor filter consists of a sinusoidal plane wave of a particular orientation and frequency, modulated by a Gaussian envelope [4]. Gabor filters are employed because they have frequency-selective and orientation-selective properties. These properties allow the filter to be tuned to give maximal response to ridges at a specific orientation and frequency in the fingerprint image. Therefore, a properly tuned Gabor filter can be used to effectively preserve the ridge structures while reducing noise.

The even-symmetric Gabor filter is the real part of the Gabor function, which is given by a cosine wave modulated by a Gaussian (see Figure 3.3). An even symmetric Gabor filter in the spatial domain is defined as:

$$G(x, y;\theta, f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right\}\cos(2\pi f x_\theta) \tag{3.14}$$

$$x_\theta = x\cos\theta + y\sin\theta \tag{3.15}$$

$$y_\theta = -x\sin\theta + y\cos\theta \tag{3.16}$$

where θ is the orientation of the Gabor filter, f is the frequency of the cosine wave, σ_x and σ_y are the standard deviations of the Gaussian envelope along the x and y axes, respectively, and x_θ and y_θ define the x and y axes of the filter coordinate frame, respectively.

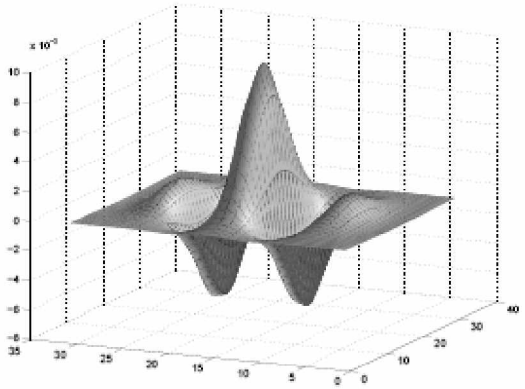


Figure 3.3: An even-symmetric Gabor filter in the spatial domain.

The Gabor filter is applied to the fingerprint image by spatially convolving the image with the filter. The convolution of a pixel (*i, j*) in the image requires the corresponding orientation value *O*(*i,j*) and ridge frequency value *F*(i, *j*) of that pixel. Hence, the application of the Gabor filter *G* to obtain the enhanced image *E* is performed as follows:

$$E(i, j) = \sum_{u=-w_x/2}^{w_x/2} \sum_{v=-w_y/2}^{w_y/2} G(u, v; O(i, j), F(i, j))N(i-u, j-v), \tag{3.17}$$

where *O* is the orientation image, *F* is the ridge frequency image, *N* is the normalized fingerprint image, and w_x and w_y are the width and height of the Gabor filter mask, respectively.

The filter bandwidth, which specifies the range of frequency the filter responds to, is determined by the standard deviation parameters σ_x and σ_y. Since the bandwidth of the filter is tuned to match the local ridge frequency, then it can be deduced that the parameter selection of σ_x and σ_y should be related with the ridge frequency. However, in the original

algorithm by Hong et al., σ_x and σ_y were empirically set to fixed values of 4.0 and 4.0, respectively.

A drawback of using fixed values is that it forces the bandwidth to be constant, which does not take into account the variation that may occur in the values of the ridge frequency. For example, if a filter with a constant bandwidth is applied to a fingerprint image that exhibits significant variation in the frequency values, it could lead to non-uniform enhancement or other enhancement artefacts. Thus, rather than using fixed values, I have chosen the values of σ_x and σ_y to be a function of the ridge frequency parameter, which are defined as:

$$\sigma_x = k_x F(i, j), \tag{3.18}$$

$$\sigma_y = k_y F(i, j), \tag{3.19}$$

where F is the ridge frequency image, k_x is a constant variable for σ_x, and k_y is a constant variable for σ_y. This allows a more adaptable approach to be used, as the values of σ_x and σ_y can now be specified adaptively according to the local ridge frequency of the fingerprint image.

Furthermore, in the original algorithm, the width and height of the filter mask were both set to fixed values of 11. The filter size controls the spatial extent of the filter, which ideally should be able to accommodate the majority of the useful Gabor waveform information. However, a fixed filter size is not optimal in that it does not allow the accommodation of Gabor waveforms of different sized bandwidths. Hence, to allow the filter size to vary according to the bandwidth of the Gabor waveform, I have set the filter size to be a function of the standard deviation parameters:

$$w_x = 6\sigma_x \tag{3.20}$$

$$w_y = 6\sigma_y \tag{3.21}$$

where w_x and w_y are the width and height of the Gabor filter mask, respectively, and σ_x and σ_y are the standard deviations of the Gaussian envelope along the x and y axes, respectively. In the above equation, the width and height of the filter mask are both specified as 6σ, due to most of the Gabor wave information being contained within the region [-3σ ; 3σ] away from the y axis. Hence, this selection of parameters allows the filter mask to capture the majority of the Gabor waveform information.

3.2.6 Binarization

Most minutiae extraction algorithms operate on binary images where there are only two levels of interest: the black pixels that represent ridges, and the white pixels that represent valleys. Binarization is the process that converts a grey level image into a binary image. This improves the contrast between the ridges and valleys in a fingerprint image, and consequently facilitates the extraction of minutiae.

One useful property of the Gabor filter is that it has a DC component of zero, which means the resulting filtered image has a mean pixel value of zero. Hence, straightforward binarization of the image can be performed using a global threshold of zero. The binarisation process involves examining the grey-level value of each pixel in the enhanced image, and, if the value is greater than the global threshold, then the pixel value is set to a binary value one; otherwise, it is set to zero. The outcome is a binary image containing two levels of information, the foreground ridges and the background valleys.

3.2.7 Thinning

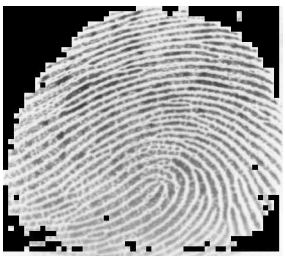
The final image enhancement step typically performed prior to minutiae extraction is thinning. Thinning is a morphological operation that successively erodes away the foreground pixels until they are one pixel wide. A standard thinning algorithm [7] is employed, which performs the thinning operation using two sub iterations. This algorithm is accessible in MATLAB via the `thin' operation under the bwmorph function. Each sub iteration begins by examining the neighbourhood of each pixel in the binary image, and based on a particular set of pixel-deletion criteria, it checks whether the pixel can be deleted or not. These sub iterations continue until no more pixels can be deleted.

The application of the thinning algorithm to a fingerprint image preserves the connectivity of the ridge structures while forming a skeletonised version of the binary image. This skeleton image is then used in the subsequent extraction of minutiae. The process involving the extraction of minutiae from a skeleton image will be discussed in the next chapter.

3.3 EXPERIMENTAL RESULTS

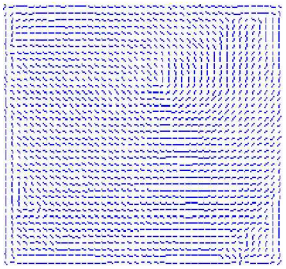
All the methods and algorithms described in this dissertation were implemented using MATLAB V7.0. When testing the performance of the enhancement algorithm, the computational time was not measured. The aim of the experimental results section is to illustrate the results of each stage in the enhancement algorithm and to assess how well each stage performs.

Original Image

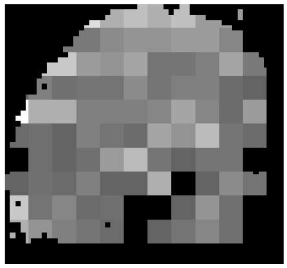


Segmented Image

Normalized Image



Orientation Image



Ridge Frequency Image

Filtered Image

Enhanced Image

Additional Enhancement Results

Input images

Output images

Chapter 4

MINUTIAE EXTRACTION AND POSTPROCESSING

4.1 INTRODUCTION

After a fingerprint image has been enhanced, the next step is to extract the minutiae from the enhanced image. Following the extraction of minutiae, a final image postprocessing stage is performed to eliminate false minutiae. This chapter provides discussion on the methodology and implementation of techniques for minutiae extraction and fingerprint image postprocessing. The last section presents the results from the experiments conducted using the implemented techniques.

4.2 METHODOLOGY

This section describes the methodology for performing the minutiae extraction and image postprocessing stages. The minutiae extraction technique based on the widely employed Crossing Number method is implemented. For the image postprocessing stage, the minutiae validation algorithm by Tico and Kuosmanen [11] is implemented. Firstly, the minutiae extraction method will be discussed, followed by details of the minutiae validation algorithm.

4.2.1 Minutiae Extraction

The Crossing Number (CN) method is used to perform minutiae extraction. This method extracts the ridge endings and bifurcations from the skeleton image by examining the local neighbourhood of each ridge pixel using a 3×3 window. The CN for a ridge pixel P is given by:

$$CN = 0.5\sum_{i=1}^{8}\left|P_i - P_{i+1}\right|, \qquad P_9 = P_1 \tag{4.1}$$

where P_i is the pixel value in the neighbourhood of P. For a pixel P, its eight neighbouring pixels are scanned in an anti-clockwise direction as follows:

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

After the CN for a ridge pixel has been computed, the pixel can then be classified according to the property of its CN value. As shown in Figure 4.1, a ridge pixel with a CN of one corresponds to a ridge ending, and a CN of three corresponds to a bifurcation.
For each extracted minutiae point, the following information is recorded:

- *x* and *y* coordinates,
- orientation of the associated ridge segment, and
- type of minutiae (ridge ending or bifurcation).

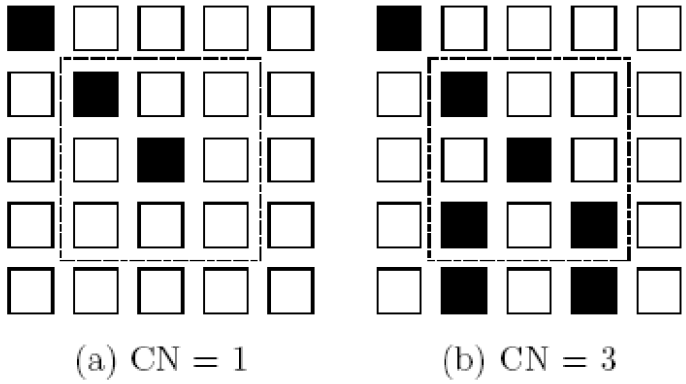


Figure 4.1: Examples of a ridge ending and bifurcation pixel. (a) A Crossing Number of one corresponds to a ridge ending pixel. (b) A Crossing Number of three corresponds to a bifurcation pixel.

4.1.2 Fingerprint Image Postprocessing

In order to eliminate false minutiae, the minutiae validation algorithm by Tico and Kuosmanen [11] tests the validity of each minutiae point by scanning the skeleton image and examining the local neighbourhood around the point. The first step in the algorithm is to create an image *M* of size $W \times W$, where *M* corresponds to the $W \times W$ neighbourhood centered on the candidate minutiae point in the skeleton image. The central pixel of *M* corresponds to the minutiae point in the skeleton image, and so this pixel is labelled with a value of -1. The rest of the pixels in *M* are initialized to values of zero, as shown in Figure 4.3(a) and Figure 4.4(a). The subsequent steps of the algorithm depend on whether the candidate minutiae point is a ridge ending or a bifurcation.

1. For a candidate ridge ending point:

(a) Firstly, label with a value of 1 all the pixels in *M*, which are eight-connected with the ridge ending point (see Figure 4.2(b)).

(b) The next step is to count in a clockwise direction, the number of 0 to 1 transitions (T_{01}) along the border of image *M*. If T_{01} = 1, then the candidate minutiae point is validated as a true ridge ending.

2. For a candidate bifurcation point:

(a) Firstly, examine the eight neighbouring pixels surrounding the bifurcation point in a clockwise direction. For the three pixels that are connected with the bifurcation point, label them with the values of 1, 2, and 3, respectively. An example of this initial labeling process is shown in Figure 4.4(b).

(b) The next step is to label the rest of the ridge pixels that are connected to these three connected pixels. This labeling is similar to the ridge ending approach; however, instead of labeling a single ridge branch, three ridge branches are now labelled. Let l = 1, 2 and 3 represent the label for each ridge branch. For each l, label with l all the ridge pixels that have a label of 0, and are connected to an l labelled pixel. Examples of the bifurcation labeling process are shown in Figures 4.4(c), (d) and (e).

(c) The last step is to count in a clockwise direction, the number of transitions from 0 to 1 (T_{01}), 0 to 2 (T_{02}), and 0 to 3 (T_{03}) along the border of image *M*. If T_{01} = 1 ^ T_{02} = 1 ^ T_{03} = 1, then the candidate minutiae point is validated as a true bifurcation.

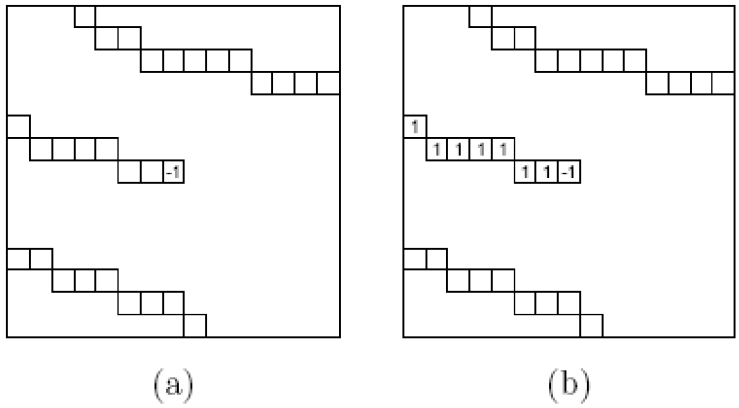


Figure 4.2: Example of validating a candidate ridge ending point. T_{01} = 1.

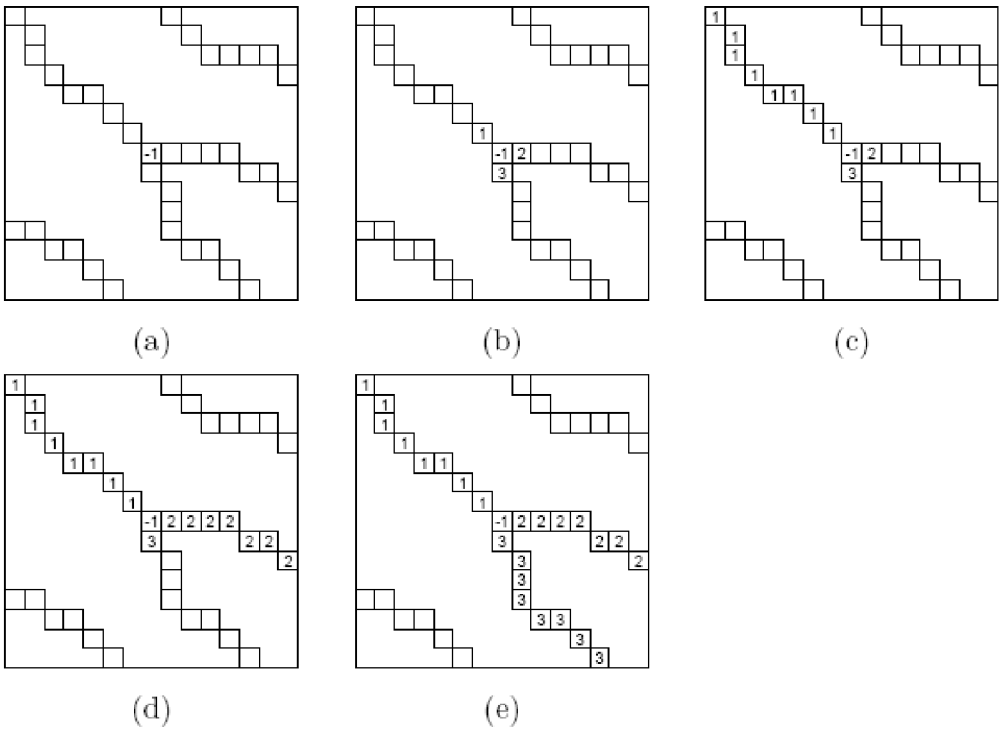
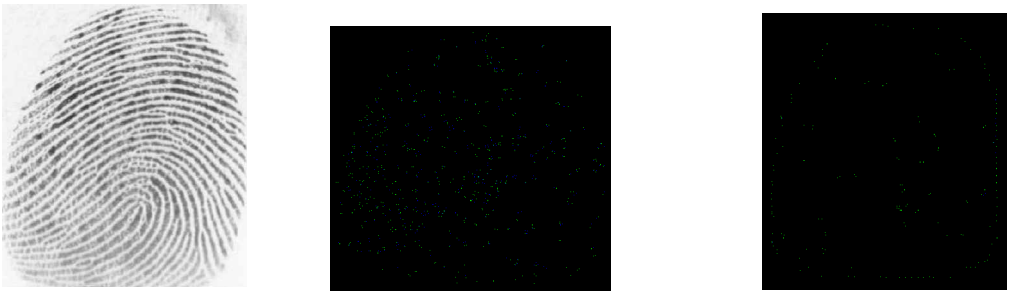


Figure 4.3: Example of validating a candidate bifurcation point. $T_{01} = 1 \wedge T_{02} = 1 \wedge T_{03} = 1$.

4.3 EXPEIMENTAL RESULTS

This section presents results of performing the minutiae extraction and image processing stages on a series of real fingerprint images. Experiments are firstly conducted to assess how well the Crossing Number (CN) technique is able to extract the minutiae from the skeleton image.



Original image minutiae before enhancement after enhancement

Minutiae points Green=ridge ending pixels
Blue=ridge bifurcations

Chapter 5

MINUTIAE MATCHING

5.1 INTRODUCTION

In this chapter, an alignment-based matching algorithm is implemented. Recognition by alignment has received a great deal of attention during the past few years, because it is simple in theory, efficient in discrimination, and fast in speed. Our alignment-based matching algorithm decomposes the minutia matching into two stages:

1) *Alignment stage*, where transformations such as translation, rotation and scaling between an input and a template in the database are estimated and the input minutiae are aligned with the template minutiae according to the estimated parameters; and

2) *Matching stage*, where both the input minutiae and the template minutiae are converted to polygons in the polar coordinate system and an elastic string matching algorithm is used to match the resulting polygons.

5.2 ALIGNMENT OF POINT PATTERNS

Ideally, two sets of planar point patterns can be aligned completely by two corresponding point pairs. A true alignment between two point patterns can be obtained by testing all possible corresponding point pairs and selecting the optimal one. However, due to the presence of noise and deformations, the input minutiae cannot always be aligned exactly with respect to those of the templates. In order to accurately recover pose transformations between two point patterns, a relatively large number of corresponding point pairs need to be used. This leads to a prohibitively large number of possible correspondences to be tested. Therefore, an alignment by corresponding point pairs is not practical even though it is feasible.

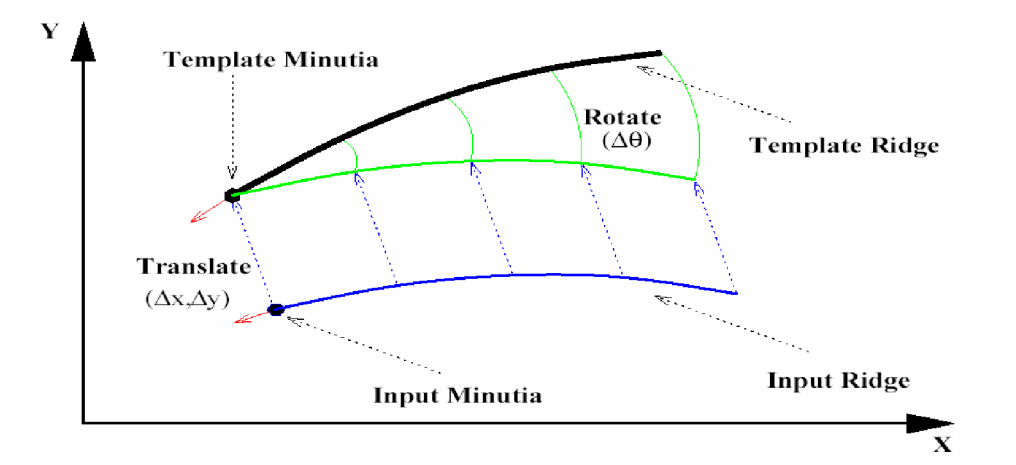


Fig. 5.1 Alignment of the input ridge and the template ridge.

It is well known that corresponding curve segments are capable of aligning two point patterns with a high accuracy in the presence of noise and deformations. Each minutia in a fingerprint is associated with a ridge. It is clear that a true alignment can be achieved by aligning corresponding ridges (see Fig. 5.1). During the minutiae detection stage, when a minutia is extracted and recorded, the ridge on which it resides is also recorded. This ridge is represented as a planar curve with its origin coincident with the minutia and its x-coordinate being in the same direction as the direction of the minutia. Also, this planar curve is normalized with the average inter-ridge distance. By matching these ridges, the relative pose transformation between the input fingerprint and the template can be accurately estimated. To be specific, let R_d and R_D denote the sets of ridges associated with the minutiae in input image and template, respectively. Our *alignment algorithm* can be described in terms of the following steps:

1) For each ridge $d \in R^d$, represent it as an one-dimensional discrete signal and match it against each ridge, $D \in R^D$ according to the following formula:

$$S = \frac{\sum_{i=0}^{L} d_i D_i}{\sqrt{\sum_{i=0}^{L} d_i^{\,2} D_i^{\,2}}} \tag{5.1}$$

where L is the minimal length of the two ridges and d_i and D_i represent the distances from point i on the ridges d and D to the x-axis, respectively. The sampling interval on a ridge is set to the average interridge distance. If the matching score S $(0 \le S \le 1)$ is larger than a certain threshold T_r, then go to step 2, otherwise continue to match the next pair of ridges.

2) Estimate the pose transformation between the two ridges (Fig. 5.1). Generally, a least-

square method can be used to estimate the pose transformation. However, in our system, we observe that the following method is capable of achieving the same accuracy with less computation. The translation vector $(\Delta x, \Delta y)^T$ between the two corresponding ridges is computed by

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} x^d \\ y^d \end{pmatrix} - \begin{pmatrix} x^D \\ y^D \end{pmatrix} \tag{5.2}$$

where $(x^d, y^d)^T$ and $(x^D, y^D)^T$ are the *x* and *y* coordinates of the two minutiae, which are called reference minutiae, associated with the ridges *d* and *D*, respectively. The rotation angle $\Delta\theta$ between the two ridges is computed by

$$\Delta\theta = \frac{1}{L}\sum_{i=0}^{L}(\gamma_i - \Gamma_i) \tag{5.3}$$

where *L* is the minimal length of the two ridges *d* and *D*; γ_i and Γ_i are radial angles of the i^{th} point on the ridge with respect to the reference minutia associated with the two ridges *d* and *D*, respectively. The scaling factor between the input and template images is assumed to be one. This is reasonable, because fingerprint images are captured with the same device in both the off-line processing phase and the on-line verification phase.

3) Denote the minutia $(x^d, y^d, \theta^d)^T$, based on which the pose transformation parameters are estimated, as the reference minutia. Translate and rotate all the *N* input minutiae with respect to this reference minutia, according to the following formula:

$$\begin{pmatrix} x_i^A \\ y_i^A \\ \theta_i^A \end{pmatrix} = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{pmatrix} + \begin{pmatrix} \cos\Delta\theta & \sin\Delta\theta & 0 \\ \sin\Delta\theta & -\cos\Delta\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i - x^d \\ y_i - y^d \\ \theta_i - \theta^d \end{pmatrix} \tag{5.4}$$

where $(x_i, y_i, \theta_i)^T$, (i = 1, 2..., N), represents an input minutia and $(x_i^A, y_i^A, \theta_i^A)^T$ represents the corresponding aligned minutia.

5.2 ALIGNED POINT PATTERN MATCHING

If two identical point patterns are exactly aligned with each other, each pair of corresponding points is completely coincident. In such a case, a point pattern matching can be simply achieved by counting the number of overlapping pairs. However, in practice, such a situation is not encountered. On the one hand, the error in determining and localizing minutia hinders the alignment algorithm to recover the relative pose transformation exactly, while on the other hand, our alignment scheme described above does not model the nonlinear deformation of fingerprints which is an inherent property of fingerprint impressions. With the existence of such a nonlinear deformation, it is impossible to exactly recover the position of each input minutia with respect to its corresponding minutia in the template. Therefore, the aligned point pattern matching algorithm needs to be elastic which means that it should be capable of tolerating, to some extent, the deformations due to inexact extraction of minutia positions and nonlinear deformations. Usually, such an elastic matching can be achieved by placing a bounding box around each template minutia, which specifies all the possible positions of the corresponding input minutia with respect to the template minutia, and restricting the corresponding minutia in the input image to be within this box [18]. This method does not provide a satisfactory performance in practice, because local deformations may be small while the accumulated global deformations can be quite large. We have implemented an adaptive elastic matching algorithm with the ability to compensate the minutia localization errors and nonlinear deformations.

Let

$$P = \left(\left(x_1^{\,P}, y_1^{\,P}, \theta_1^{\,P} \right)^T, \mathsf{h}, \left(x_M^{\,P}, y_M^{\,P}, \theta_M^{\,P} \right)^T \right)$$

denote the set of M minutiae in the template and

$$Q = \left(\left(x_1^{\,Q}, y_1^{\,Q}, \theta_1^{\,Q} \right)^T, \mathsf{h}, \left(x_N^{\,Q}, y_N^{\,Q}, \theta_N^{\,Q} \right)^T \right)$$

denote the set of N minutiae in the input image which is aligned with the above template with respect to a given reference minutia point. The steps in our elastic point pattern matching algorithm are given below:

1) Convert each minutia point to the polar coordinate system with respect to the corresponding reference minutia on which the alignment is performed:

$$\begin{pmatrix} r_i \\ e_i \\ \theta_i \end{pmatrix} = \begin{pmatrix} \sqrt{(x_i^* - x^r)^2 + (y_i^* - y^r)^2} \\ \tan^{-1}\left(\dfrac{y_i^* - y^r}{x_i^* - x^r}\right) \\ \theta_i^* - \theta^r \end{pmatrix} \tag{5.6}$$

where $\left(x_i^*, y_i^*, \theta_i^*\right)$ are the coordinates of a minutia, $\left(x^r, y^r, \theta^r\right)^T$ are the coordinates of the reference minutia, and $\left(r_i, e_i, \theta_i\right)^T$ is the representation of the minutia in polar coordinate system (r_i represents the radial distance, e_i represents the radial angle, and q_i represents the orientation of the minutia with respect to the reference minutia).

2) Represent the template and the input minutiae in the polar coordinate system as symbolic strings by concatenating each minutia in the increasing order of radial angles:

$$P_P = \left(\left(r_1^P, e_1^P, \theta_1^P\right)^T, ..., \left(r_M^P, e_M^P, \theta_M^P\right)^T\right) \tag{5.7}$$

$$Q_P = \left(\left(r_1^Q, e_1^Q, \theta_1^Q\right)^T, ..., \left(r_N^Q, e_N^Q, \theta_N^Q\right)^T\right) \tag{5.8}$$

where $\left(r_*^P, e_*^P, \theta_*^P\right)^T$ and $\left(r_*^Q, e_*^Q, \theta_*^Q\right)^T$ represent the corresponding radius, radial angle, and normalized minutia orientation with respect to the reference minutia, respectively.

3) Match the resulting strings P_p and Q_p with a dynamic programming algorithm [4] to find the edit distance between P_p and Q_p which is described below.

4) Use the edit distance between P_p and Q_p to establish the correspondence of the minutiae between P_p *and* Q_p. The matching score, M_{pq}, is then computed according to the following formula:

$$M_{PQ} = \frac{100 N_{pair}}{\max\{M, N\}} \tag{5.9}$$

where N_{pair} is the number of the minutiae which fall in the bounding boxes of template minutiae. The maximum and minimum values of the matching score are 100 and 1, respectively. The former value indicates a perfect match, while the later value indicates no match at all.

Minutia matching in the polar coordinate has several advantages. We have observed that the nonlinear deformation of fingerprints has a radial property. In other words, the nonlinear deformation in a fingerprint impression usually starts from a certain point (region) and nonlinearly radiates outward. Therefore, it is beneficial to model it in the polar space. At the same time, it is much easier to formulate rotation, which constitutes the main part of the alignment error between an input image and a template, in the polar space than in the Cartesian space. The symbolic string generated by concatenating points in an increasing order of radial angle in polar coordinate uniquely represents a point pattern. This reveals that the point pattern matching can be achieved with a string matching algorithm.

A number of string matching algorithms have been reported in the literature [4]. Here, we are interested in incorporating an elastic criteria into a string matching algorithm. Generally, string matching can be thought of as the maximization/minimization of a certain cost function such as the edit distance. Intuitively, including an elastic term in the cost function of a string matching algorithm can achieve a certain amount of error tolerance. Given two strings P_p and Q_p of lengths M and N, respectively, the edit distance, $C(M, N)$, in our algorithm is recursively defined with the following equations:

$$C(m,n) = \begin{cases} 0 & if \quad m = 0, \quad or \quad n = 0 \\ \min \begin{cases} C(m-1,n) + \Omega \\ C(m,n-1) + \Omega \\ C(m-1,n-1) + w(m,n) \end{cases} \end{cases} \quad 0 < m \le M, and\ 0 < n \le N \tag{5.10}$$

$$w(m,n) = \begin{cases} \alpha \left| r_m^P - r_n^Q \right| + \beta \Delta e + \gamma \Delta\theta & if \left| r_m^P - r_n^Q \right| < \delta, \\ & \Delta e < \varepsilon \ and \ \Delta\theta < \varepsilon \\ \Omega & otherwise \end{cases} \tag{5.11}$$

$$\Delta e = \begin{cases} a & if \left(a = \left(e_m^P - e_n^Q + 360 \right) \bmod 360 \right) < 180 \\ a - 180 & otherwise \end{cases} \tag{5.12}$$

$$\Delta\theta = \begin{cases} a & if \left(a = \left(\theta_m^P - \theta_n^Q + 360 \right) \bmod 360 \right) < 180 \\ a - 180 & otherwise \end{cases} \tag{5.13}$$

where α, β, and γ are the weights associated with each component, respectively; δ, ε, and $\in$ specify the bounding box; and Ω is a pre-specified penalty for a mismatch. Such an edit distance, to some extent, captures the elastic property of string matching. It represents a cost of changing one polygon to the other. However, this scheme can only tolerate, but not compensate for, the adverse effect on matching produced by the inexact localization of minutia and nonlinear deformations. Therefore, an adaptive mechanism is needed. This adaptive mechanism should be able to track the local nonlinear deformation and inexact alignment and try to alleviate them during the minimization process. However, we do not expect that this adaptive mechanism can handle the "order flip" of minutiae, which, to some extent, can be solved by an exhaustive reordering and matching within a local angular window.

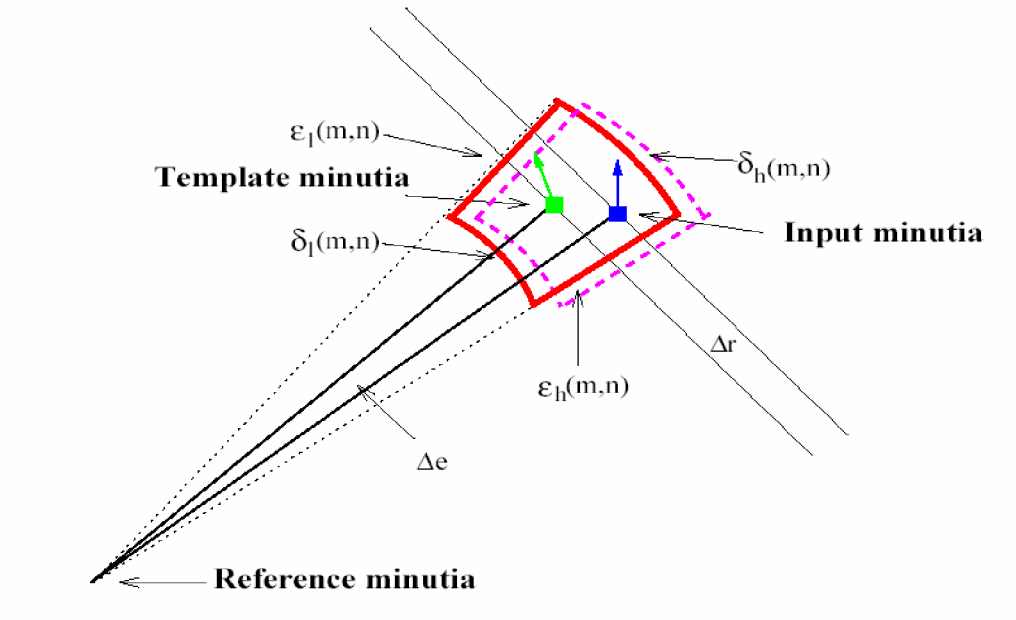


Fig 5.2: Bounding box and its adjustment

In our matching algorithm, the adaptation is achieved by adjusting the bounding box (Fig. 5.2) when an inexact match is found during the matching process. It can be represented as follows:

$$W' = \begin{cases} \alpha \left| r_m^P - r_n^Q \right| + \beta \Delta e + \gamma \Delta \theta & if \begin{cases} \delta_1(m,n) \\ < \left(r_m^P - r_n^Q \right) \\ < \delta_b(m,n) \\ \\ \varepsilon_1(m,n) \\ < \Delta e \\ < \varepsilon_b(m,n) \\ \Delta\theta < \varepsilon \end{cases} \\ \Omega & otherwise \end{cases} \tag{5.14}$$

$$\begin{pmatrix} \Delta r_a \\ \Delta e_a \end{pmatrix} = \begin{cases} \begin{pmatrix} r_m^P - r_n^Q \\ \Delta e \end{pmatrix} & if \begin{cases} \delta_1(m,n) \\ < \left(r_m^P - r_n^Q \right) \\ < \delta_b(m,n) \\ \\ \varepsilon_1(m,n) \\ < \Delta e \\ < \varepsilon_b(m,n) \\ \Delta\theta < \varepsilon \end{cases} \\ 0 & otherwise \end{cases} \tag{5.15}$$

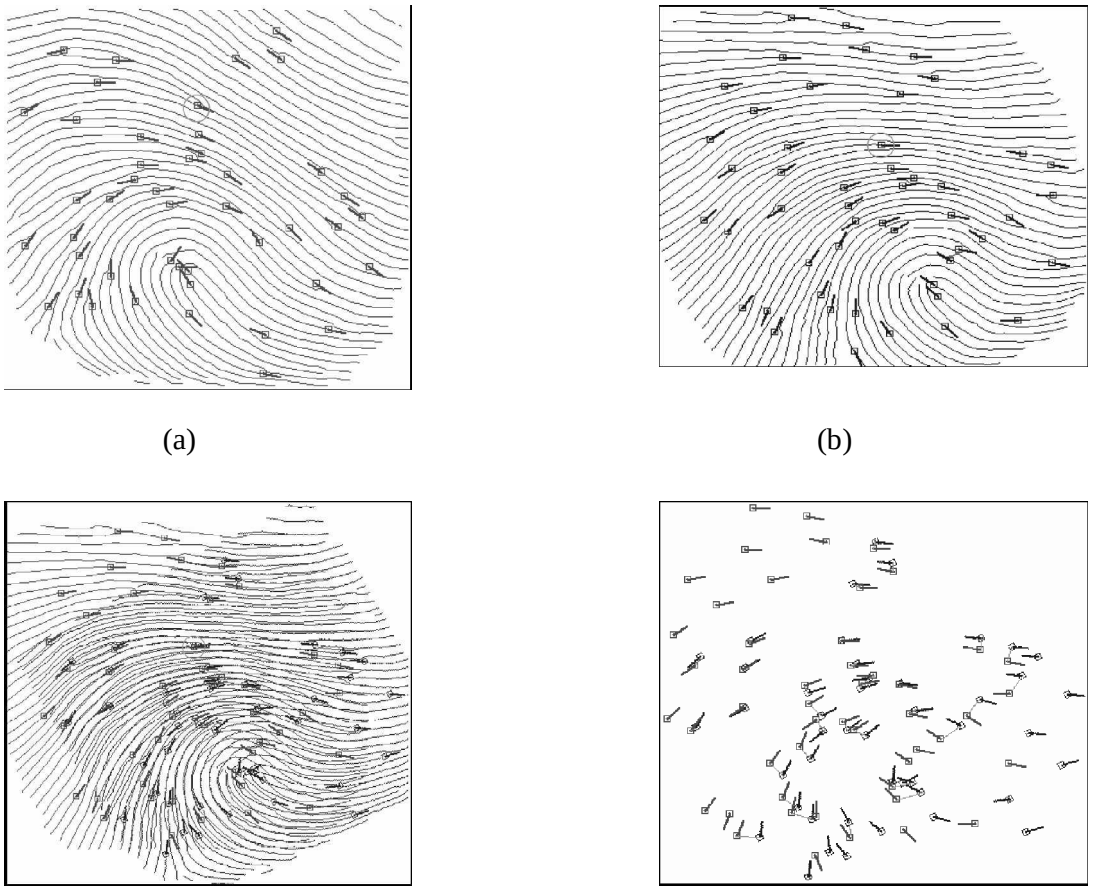
$$\delta_1(m+1,n+1) = \delta_1(m,n) + \eta \Delta r_a \tag{5.16}$$

$$\delta_h(m+1,n+1) = \delta_h(m,n) + \eta \Delta r_a \tag{5.17}$$

$$e_1(m+1,n+1) = e_1(m,n) + \eta \Delta e_a \tag{5.18}$$

$$e_h(m+1,n+1) = e_h(m,n) + \eta \Delta e_a \tag{5.19}$$

where $w'(m, n)$ represents the penalty for matching a pair of minutiae $\left(r_m^{\ P}, e_m^{\ P}, \theta_m^{\ P}\right)^T$ and $\left(r_n^{\ Q}, e_n^{\ Q}, \theta_n^{\ Q}\right)^T$, $\delta_l(m,n)$, $\delta_h(m,n)$, $\varepsilon_l(m,n)$, and $\varepsilon_h(m,n)$ specify the adaptive bounding box in the polar coordinate system (radius and radial angle); and η is the learning rate. This elastic string matching algorithm has a number of parameters which are critical to its performance. We have empirically determined the values of these parameters as follows: $\delta_l(0, 0) = -8$; $\delta_h(0, 0) = +8$; $\varepsilon_l(0, 0) = -7.5$; $\varepsilon_h(0, 0) = +7.5$; $\varepsilon = 30$; $\alpha = 1.0$; $\beta = 2.0$; $\gamma = 0.1$; $\Omega = 200(\alpha + \beta + \gamma)$; $\eta = 0.5$. The values of δ_l (0, 0), δ_h (0, 0), $\varepsilon_l(0, 0)$, and $\varepsilon_h(0, 0)$ depend on the resolution of fingerprint images. Fig. 5.3 shows the results of applying the matching algorithm to an input minutia set and a template.



(a) (b)

(c) (d)

Fig 5.3: Results of applying the matching algorithm to an input minutiae set and a template. (a) Input minutiae set. (b) Template minutiae set. (c) Alignment result based on the minutiae marked with green circles. (d) Matching result where template minutiae and their correspondences are connected by green lines.

Chapter 6

APPLYING DWT AND DCT FEATURES FOR FINGERPRINT MATCHING

6.1 INTRODUCTION

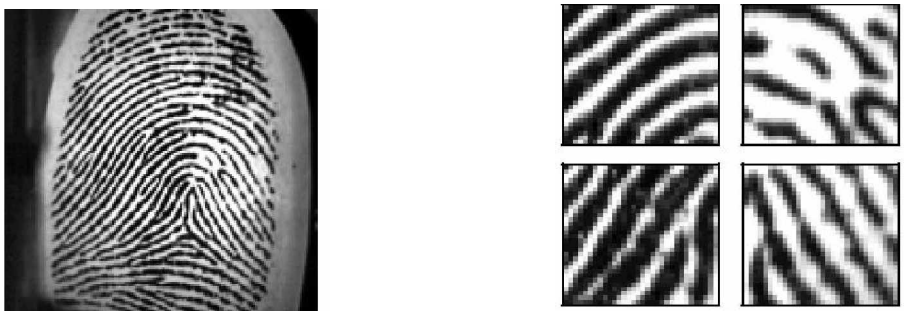
Basically, methods of fingerprint matching are categorized in three classes [2] i.e. correlation-based matching, minutiae-based matching and ridge feature based matching. Correlation-based matching is a straight method used to compare the corresponding pixels of the fingerprint images in many of rotating and shifting. In contrary, minutiae-based matching methods, e.g. [17, 18], extract some informative features known as minutiae from the fingerprint images, and use them for fingerprint matching purposes. The latter methods usually require some pre-processing processes, namely binarization and thinning. The objectives of such pre-processing processes are image normalization and background noise elimination. That is, binarization is used to change a gray-scale image to black-and-white one, while thinning is used to change a thick ridge line to the one that has only one pixel width. In ridge feature-based matching, the feature of ridge was extracted from the gray-scale images e.g. a GABOR filter based method [19,20]. In their method, the ridge features were directly extracted from the fingerprint image in 4 orientations (0° , 450 , 90° and 1450), and used for fingerprint matching. However, it was later shown that a higher recognition rate can be achieved with the similar method using wavelet features [21].

In this chapter, Discrete Cosine Transform (DCT) is considered and used to extract such informative features for fingerprint matching purpose. The performance in terms of recognition rate, and processing time required for features extracting and matching of our approach is evaluated and compared to the most efficient DWT based method proposed to date [21].

6.2 EXTRACTION METHOD FOR DWT FEATURES

To extract the DWT features from a fingerprint, a gray-scale fingerprint image is cropped to the size of 64x64 pixels, where the center or core point in the image is referred to as a reference point (see Fig. 6.1a). The cropped image is then quartered, centered at the reference point, to obtain four non-overlapping images of size 32x32 pixels. Apparently, a fingerprint image is merely a redundancy of white and black lines (see Fig.6.lb), which normally causes

oscillatory patterns in the middle frequency channels. The energy distribution over the middle scales in the frequency domain can hence be considered as an informative criterion for fingerprint pattern classification.



(a) (b)

Fig.6.1.(a) a fingerprint image (b) four 32 x 32-pixel images, cropped and quartered at its centre

With the DWT, a two-dimensional wavelet decomposition on J octaves of a discrete image ao [n, m] can represent that image as 3J+1 sub-images

$$\left[a_{j,} \left\{ d_j^1, d_j^2, d_j^3 \right\}_{j=1,\ldots,J} \right] \qquad (6.1)$$

where a_j is a low resolution approximation of the original image, and d, are the wavelet sub-images containing the image details at different scales (2') and orientations (k) i.e. at horizontal, vertical and both orientations. It is shown in [6] that the standard deviation of the DWT coefficients from each wavelet sub-image could be computed to create a feature vector of the length 3 J

$$\left[\left\{ \sigma_j^1, \sigma_j^2, \sigma_j^3 \right\}_{j=1,\ldots,J} \right] \qquad (6.2)$$

and used as a fingerprint matching parameter. The results, on J = 4 and a feature vector of length 12 (48 in total), showed significant improvements over the previous methods based on the features computed from the norm of each wavelet sub-image [7] and the GABOR filters [5]. Twelve sub-images after four time wavelet transforms are illustrated in Fig. 6.2. The performance indicator they used was the recognition rate obtained from the k-nearest neighbors (k-NN) classifier with no rejection option. Recall that, in k-NN classifier, the

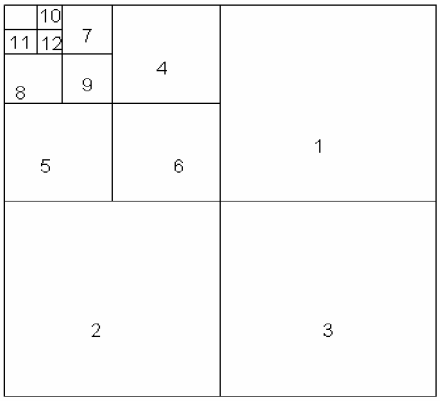


Fig. 6.2 Arrangements of twelve wavelet sub-images for feature extraction

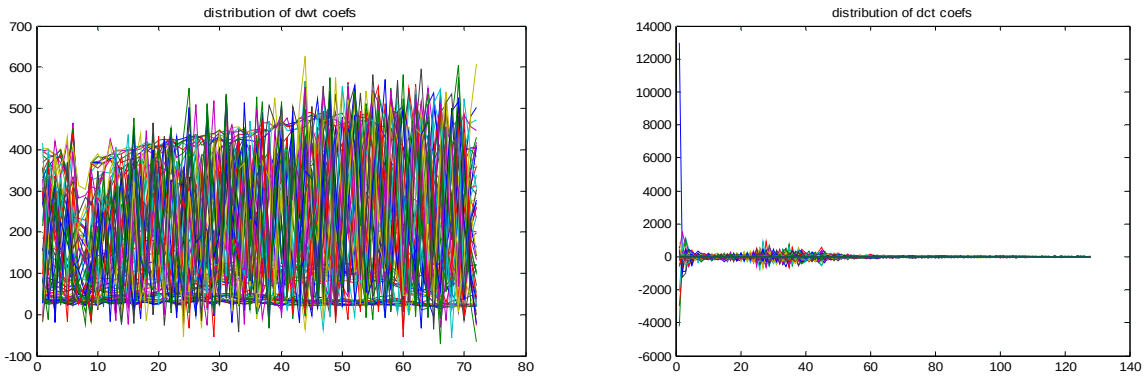
database is divided into 2 data sets, namely training set and testing set. Basically, training set is a set of fingerprint images used as a fingerprint image database, while testing set is a set of testing fingerprint images. Hence, k nearest neighbor is merely k instances in the training set which nearest to the testing set and nearest neighbor is evaluated by the distance between both sets. The most widely used distance metric is the Euclidean measure which described as follows

$$d = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{M} (A_{i,j} - B_{i,j})^2} \qquad (6.3)$$

where d was the distance between two 2-dimensional vectors A and B which their size was NxM.

6.3 EXTRACTION METHOD FOR DCT FEATURES

To observe impacts of transformation types on the informative features within a fingerprint image, the top-left part of the fingerprint shown in Fig.lb is transformed by the DWT and the DCT. The coefficients distribution in frequency domain obtained from both transforms are illustrated in Fig. 6.3. From the figure, we can see the DCT coefficients have more variation than the DWT ones, which provide a higher resolution for the fingerprint features matching. Moreover, the DCT coefficient values at middle-high frequency channels are smaller the DWT one, which generate less noises in the fingerprint matching process. In other words, informative features derived from the DCT coefficients are more distinguishable than that from the DWT coefficients.



(a) (b)

Fig. 6.3 Distribution of (a) DWT and (b) DCT coefficients.

To implement the results from our observation with the traditional features extraction process, we just replace the DWT with the DCT directly in such process. In fact, we only need to perform the DCT once, and focus on the standard deviations derived from 9 non-overlapping regions, instead of 12. That is, after the DCT is applied to the top-left part of Fig.lb, the DCT coefficients within 9 regions (no. 1-9 in Fig. 2) are calculated to obtain a feature vector of length 9 (36 in total), shown in Fig. 6.4 .

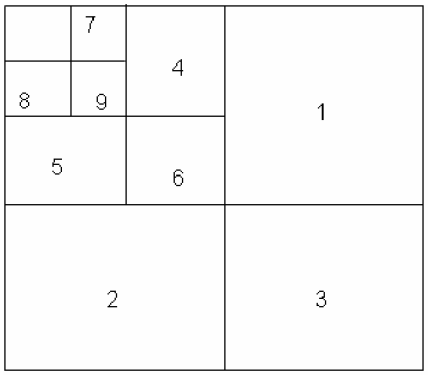


Fig 6.4 Arrangements of nine regions of DCT coefficients for feature extraction.

6.4 EXPERIMENTAL RESULTS

In the experiments, we used the fingerprint image database [23], which contains 104 fingerprint images of size 256x256 pixels including 8 images per finger from 13 individuals. As mentioned earlier, the k-NN classification with no rejection option was used as a performance indicator, where k images per individual were selected as the training set and the remaining 8-k images were used as the testing set (see Table 6.1). Determining a core point in each fingerprint image and marking it as a reference were achieved manually to obtain a precise reference point. The images were cropped to 64x64 pixels region, centered at the reference point, and then quartered to obtain four nonoverlapping images of size 32x32 pixels. The DCT method was tested, and its performance was compared with the beat results obtained from [21] using Daubechies wavelet filter (db) with 8 and 9 vanishing moments and Symmlet orthonormal wavelet filters (sym) with 5 and 9 vanishing moments, respectively. The resulting k-NN recognition rates, as illustrated in Table 6.2, obviously indicated superior performance of the DCT method.

Table 6.1: No. of Fingerprint images in Training and Testing Sets

k-NN	Training Set	Testing Set
1-NN	1*13=13	7*13=91
2-NN	2*13=26	6*13=78
3-NN	3*13=39	5*13=65
4-NN	4*13=52	4*13=52

Table 6.2: Recognition Rates at various k-NN Classifiers

Method	1-NN (%)	2-NN (%)	3-NN (%)	4-NN (%)
DCT	78.29	80.12	83.43	85.14
db8	56.15	60.05	61.46	62.92
db9	66.73	66.97	71.26	74.53
sym5	59.28	61.65	62.30	63.51
sym9	65.37	65.91	70.08	72.96

From the results, it can be clearly seen that the recognition rates of the DCT method were on average much higher than that of the method using the wavelet features. This is because the pattern of the DCT coefficients was more oscillated and distinctive than that of the DWT coefficients. It is worth noting that the DCT coefficients were divided into 9 regions and one standard deviation was used to represent the local property in each region,

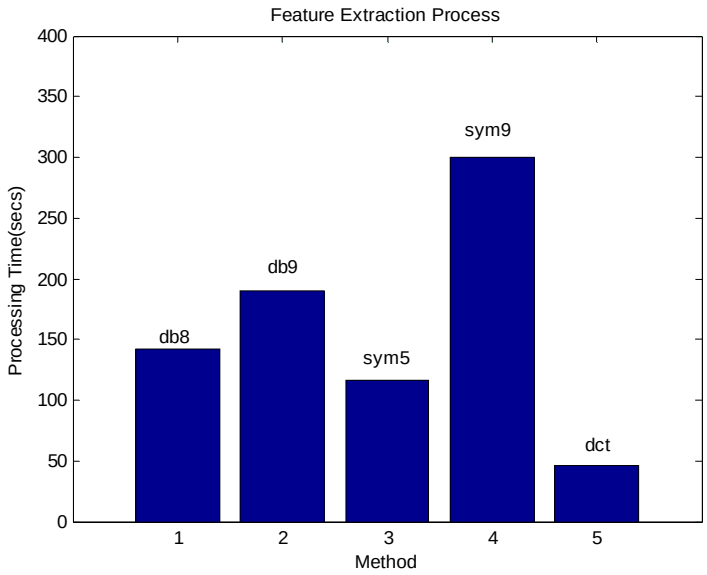


Fig. 6.5 Comparison of the processing time required in the features extraction process.

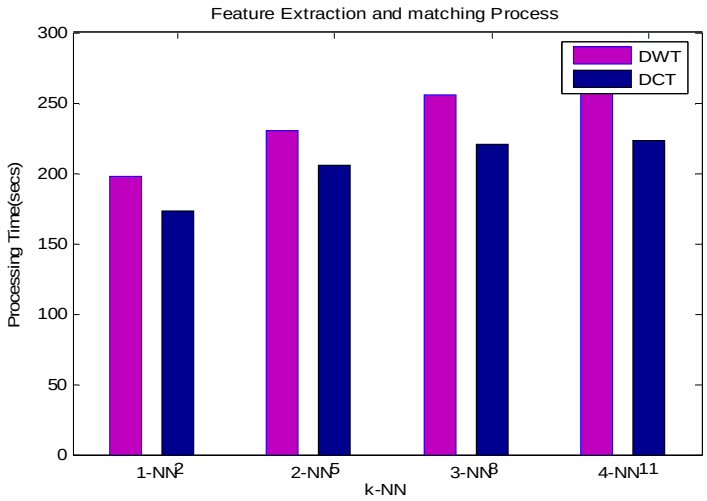


Fig. 6.6 Comparison of the processing time required in the features searching & matching processes.

while the DWT coefficients were divided into 12 sub-images and one standard deviation was used to represent the entire property in each sub-image. This would make the fingerprint features acquired from the DCT coefficients robust against erroneous fingerprint in the scanning process. In contrary, the features acquired from the DWT coefficients were entirely affected by the erroneous fingerprint scanning, especially at a coarser scale e.g. J = 4. For the feature extraction process, the complexity of the DCT method is apparently lower since the DCT is required only once, while in the previous methods, four times of DWT are

required. Furthermore, the complexity of the fingerprint searching & matching processes is also lower because of a shorter length of feature vectors, stored and used in the classification. The experimental results in term of average processing time required in the features extraction and searching & matching processes are compared and presented in Fig. 6.5 and 6.6, respectively.

Chapter 7

CONCLUSION AND FUTURE WORK

This chapter gives a summary of the work presented in this thesis. An outline for the future work based on this is also given.

6.1 CONCLUSION

The work in this thesis is mainly on the enhancement, extraction and matching of fingerprint images. Initially the spatial domain method was used for fingerprint matching and extraction. In this, the technique of fingerprint image enhancement using Gabor filter is implemented to facilitate the extraction of minutiae. The experimental results have shown that with an accurate estimation of the orientation and ridge frequency, the Gabor filter is able to effectively enhance the clarity of the ridge structures with reducing noise. In contrast, for low quality images that exhibit high intensities of noise, the filter is less effective in enhancing the image due to inaccurate estimation of the orientation and ridge frequency parameters. Overall, the results have shown that the implemented enhancement algorithm is a useful step to employ prior to minutiae extraction.

The Crossing Number method was then implemented to perform extraction of minutiae. Extensive simulations were done which show that this method is able to detect accurately all valid bifurcations and ridge endings from the thinned image. However, there are cases where the extracted minutiae do not correspond to true minutia points. Hence, an image postprocessing stage is implemented to validate the minutiae.

For matching purpose an alignment-based matching algorithm is presented. In this, an input minutiae are aligned with the template by estimating the translation, rotation and scaling parameters between an input and a template. Then the input and template minutiae are converted to polygons in the polar coordinate system and an *elastic string matching* algorithm is used to match the resulting polygons.The input which satisfy the matching score is declared as a matched fingerprint with the template.

The fingerprint extraction and matching was also tested using transform domain techniques. For this purpose DWT and DCT were used. It was found that the DCT is better suited than DWT in extracting informative features that exist in the middle frequency channels of a fingerprint image. The experimental results have shown that with DCT method, a higher recognition rate together with a lower complexity could be achieved in a fingerprint matching system, compared to the method using wavelet features.

6.2 SCOPE OF FUTURE WORK

The following are the some of the interesting extensions of the present work:

1) An investigation into a filter whose primary aim is to specifically enhance the minutia points. This project has followed the approach adopted by most previous work where the emphasis is on enhancing the ridge structures using Gabor, or Gabor-like filters. However, while the ridge structures are enhanced, this approach has shown to be less effective in enhancing areas containing minutiae points, which are the points of main interest.

2) The current minutiae verification algorithm is applied on the minutiae extracted using the algorithm that detects the minutiae in the thinned binarized fingerprint ridges. The minutiae patterns that are learnt during the training can be used to detect the minutiae in the gray scale fingerprint image directly.

3) The algorithms suggested in the thesis can also be implemented using neural networks and fuzzy logic techniques. It is also possible to compare and analyze the algorithms by using neural network.

BIBLIOGRAPHY

[1] Ruud Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior. *Guide to Biometrics*. Springer Verlag, 2003.

[2] D. Maio, D. Maltoni, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer Verlag, 2003.

[3] S. Pankanti, S. Prabhakar, and A. K. Jain, "On the individuality of fingerprints," *IEEE Transactions on PAMI*, 24(8):1010–1025, 2002.

[4] Hong, L., Wan, Y., and Jain, A. K, "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 8 (1998), 777–789.

[5] Jain, A., Hong, L., Pankanti, S., and Bolle, R, "An identity authentication system using fingerprints," In *Proceedings of the IEEE* (September 1997), vol. 85, pp. 1365–1388.

[6] Ratha, N., Chen, S., and Jain, A, "Adaptive flow orientation based feature extraction in fingerprint images," *Pattern Recognition 28*, 11 (1995), 1657–1672.

[7] Prabhakar, S., Wang, J., Jain, A. K., Pankanti, S., and Bolle, R, "Minutiae verification and classification for fingerprint matching," In *Proc. 15th International Conference Pattern Recognition (ICPR)* (September 2000), vol. 1, pp. 25–29.

[8] Jain, A. K., Prabhakar, S., and Hong, L, "A multichannel approach to fingerprint classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 4 (1999), 348–359.

[9] Sherlock, D. B. G., Monro, D. M., and Millard, K, "Fingerprint enhancement by directional Fourier filtering," In *IEE Proc. Vis. Image Signal Processing* (1994), vol. 141, pp. 87–94.

[10] Amengual, J. C., Juan, A., Prez, J. C., Prat, F., Sez, S., and Vilar, J. M., “Real-time minutiae extraction in fingerprint images,” In *Proc. of the 6th Int. Conf. on Image Processing and its Applications* (July 1997), pp. 871–875.

[11] Tico, M., and Kuosmanen, P, “An algorithm for fingerprint image postprocessing,” In *Proceedings of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers* (November 2000), vol. 2, pp. 1735–1739.

[12] J.P.P. Starink and E. Backer, “Finding Point Correspondence Using Simulated Annealing,” *Pattern Recognition*, vol. 28, no. 2, pp. 231-240, 1995.

[13] A. Ranade and A Rosenfeld, “Point Pattern Matching by Relaxation,” *Pattern Recognition*, vol. 12, no. 2, pp. 269-275, 1993.

[14] K. Karu and A.K. Jain, “Fingerprint Registration,” Research Report, Michigan State Univ., Dept. of Computer Science, 1995.

[15] K. Karu and A.K. Jain, “Fingerprint Classification,” *Pattern Recognition*, vol. 29, no. 3, pp. 389-404, 1996.

[16] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*. New York: McGraw-Hill, 1990.

[17] D.S. Zorita, J.O. Garcia, S.C. Llanas and J.G. Rodriguez, "Minutiae Extraction Scheme for Fingerprint Recognition Systems", Proceedings of the 7th International Conference on Image Processing, Vol. 3, pp. 254-257, 2001.

[18] M. Arantes, A.N. Ide and J.H. Saito, "A System for Fingerprint Minutiae Feature Classification and Recognition", Proceedings of the 9th International Conference on Image Processing, Vol. 5, pp. 2474-2478, 2002.

[19] C.J. Lee and S.D. Wang, "A GABOR Filter-Based Approach to Fingerprint Recognition", IEEE Workshop on Signal Processing Systems, pp. 371-378, 1999.

[20] C.J. Lee and S.D. Wang, "Fingerprint Features Extraction Using GABOR Filters", Electronics Letters, Vol. 35, No. 4, pp. 288-290, 1999.

[21] M. Tico, E. Immonen, P. Ramo, P. Kuosmanen and J. Saarinen, "Fingerprint Recognition Using Wavelet Features", The 2001 IEEE International Symposium on Circuits and Systems, Vol. 2, pp. 21-24, 2001.

[22] M. Tico, P. Kuosmanen, and J. Saarinen, 'Wavelet Domain Features for Method Fingerprint Recognition', Electronics Letters, Vol. 37, No. 1, pp.21-22, 2001.

[23] Biometric System Lab., University of Bologna, Cesena-Italy. (www.csr.unibo.it/research/biolab/)

[24] T. Jea, V. K. Chavan, V. Govindaraju, and J. K. Schneider. Security and matching of partial fingerprint recognition systems. In *Proceeding of SPIE*, number 5404, pages 39–50, 2004.

[25] [11] Jain, A. K., Hong, L., and Bolle, R. M. On-line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 4 (1997), 302–314.

BIOGRAPHICAL SKETCH

K.Ayyanna was born on August 27th, 1983 in Kurnool, Andhra Pradesh, India. He completed his undergraduate studies in April 2005 and received his B.Tech. degree in Electronics and Communication Engineering from G. Pulla Reddy Engineering College, Sri Krishnadevaraya University in Kurnool, India. He was admitted to Master's program at National Institute of Technology, Rourkela in the department of Electronics and Communication Engineering in July 2005. His areas of interest include Digital Signal Processing, Digital Image Processing, and Pattern Recognition. He post-graduated from National Institute of Technology, Rourkela in May 2007.